VB.NET 2008 [1]



إعداد وإخراج

أ/ زينب المخلافي

مقدمة وحل المشكلة

في هذه الوحدة نعرض مقدمة عن ماهية برنامج الحاسب ولغة البرمجة وأنواع لغات البرمجة وأهمية مهنة البرمجة، ثم بعد ذلك نشرح القواعد التي تساعد في تحليل المشكلة ومعرفة عناصرها المكونة لها و كيف يمكن تجزئة المشكلة إلى أجزاء صغيرة يسهل التعامل معها، وفيها أيضا نوضح رموز رسم خرائط التدفق ثم رسم هذه الخرائط للمشكلة بعد كتابة الخوارزم والتي تعطي صورة لحل المشكلة.

برنامج الحاسب

البرنامج هو عبارة عن مجموعة من التعليمات تعطى للحاسب للقيام بعمل ما مثل حساب مجموع قيم مختلفة، حساب المتوسط الحسابي، حساب مضروب عدد معين ....الخ
والبرنامج هو الذي يحدد للحاسب كيفية التعامل مع البيانات للحصول على النتائج المطلوبة. والبرنامج يكتب بواسطة المبرمج (Computer Programmer) الذي يفهم المشكلة ويقترح الحل وينفذه لحل هذه المشكلة ويجب أن يكون البرنامج في مجموعه صحيحاً وواضحا وليس فيه لبس أو غموض.

والبرمجيات (Software) هي التي تسهل للمستخدم استخدام المكونات المادية (Hardware) بكفاءة وراحة ويمكن تقسيم البرمجيات إلى ثلاثة أنواع رئيسية وهي: -

١ - **برامج التشغيل** Operating System

مثل النوافذ (windows) و Dos، Unix، Linux، VMS وغيرها. وهي عبارة عن برامج تقوم بدور الوسيط بين المستخدم والمكونات المادية وهي تمكن المستخدم من استخدام المكونات المادية للحاسب بكفاءة وبراحة، كما أنها تساعد المستخدم في إنشاء نظام الملفات وغيرها. ومن برامج التشغيل ما يصلح للعمل في الشبكات مثل Unix، Windows، ومنها الذي يستخدم مع الحاسب فقط مثل Dos.

٢ - **برامج التطبيقات** Application Programs

وهي برامج تساعد في إنشاء كثير من التطبيقات مثل إنشاء قاعدة بيانات والرسم باستخدام الحاسب و غيرها ومن أمثلة هذه البرامج: -

برنامج الأوتوكاد Autocad– الاكسيل Excel - الأكسس Access - الأوراكلOracle - الفوتوشوب Fotoshop وغيرها كثير.

٣ - **لغات البرمجة** Programming Languages

وهذه اللغات هي التي تستخدم في بناء البرامج المختلفة وهي تتراوح من اللغات التي تتعامل مباشرة مع المكونات المادية للحاسب والأخرى التي تتطلب تحويلها من صورتها التي تكتب بها إلى صورة أخرى يستطيع الحاسب التعامل معها.

ويوجد العديد من لغات البرمجة المستخدمة اليوم وهذه اللغات يمكن تقسيمها إلى ثلاث أنواع رئيسية هي: -

١ - لغة الآلة **Machine languages**
٢ - لغات التجميع **Assembly languages**
٣ - لغات المستوى العالي **High level languages**

لغة الآلة Machine Language

وهي اللغة الوحيدة التي يفهمها الحاسب ويستطيع التعامل معها. وهذه اللغة تعتبر لغة خاصة لكل حاسب وقد تختلف من حاسب إلى آخر وهي تعتمد على المكونات المادية للحاسب نفسه، ولغة الآلة تتكون من مجموعة أرقام من بين 1،0 التي تعطي تعليمات للحاسب للقيام بمعظم العمليات الأساسية واحدة بعد الأخرى، وهي تختلف من حاسب إلى حاسب آخر ولذلك فإننا نجد أن نفس البرنامج الذي يعمل على حاسب معين قد لا يعمل على حاسب آخر يختلف عنه في المكونات المادية. و لغة الآلة من اللغات الصعبة في التعلم للإنسان حتى بالنسبة للمبرمجين لأنها عبارة عن مجموعة من الأرقام (٠، ١) فقط. وللتغلب على هذه الصعوبة تم اقتراح لغة أخرى تعتمد على استخدام اختصارات معبرة من اللغة الإنجليزية للتعبير عن العمليات الأولية التي يقوم بها الحاسب وهذه اللغة هي لغة التجميع.

لغة التجميع Assembly Languages

هي لغة تستخدم اختصارات معبرة من اللغة الإنجليزية لتعبر بها عن العمليات الأولية التي يقوم بها الحاسب مثل إضافة Add و حفظ Store وطرح Sub وغيرها.

مثال على ذلك

```
Load    A
Add     B
Store   C
```

ونظراً لأن هذه اللغة تستخدم كلمات مختصرة من اللغة الإنجليزية فإنها تحتاج محولاً لكي يحولها إلى لغة الآلة وهو ما يسمى المجمع assembler الذي يقوم بتحويل لغة التجميع إلى لغة الآلة كي يفهمها الحاسب ويستطيع تنفيذها، وبالرغم من تقليل المجهود الملقى على عاتق المبرمج للقيام بعملية البرمجة إلا أنه ما زالت توجد مشقة عند حل أبسط المسائل لأن ذلك يتطلب معرفة وكتابة العديد من التعليمات، وهذا ما دفع المبرمجين للتفكير في لغات أخرى تقلل المجهود الكبير اللازم لكتابة الكثير من التعليمات فكانت لغات البرمجة ذات المستوى العالي.

لغات البرمجة ذات المستوى العالي High Level Languages

وهذه اللغات كتبت بحيث تستخدم بعض الكلمات الإنجليزية العادية بنفس معانيها حيث يقوم كل أمر منها بتنفيذ العديد من الواجبات، وهذه اللغات كسابقتها تحتاج إلى مترجمات Compilers التي تقوم بتحويل التعليمات(الأوامر) إلى لغة الآلة، وهذه اللغات تستخدم العلاقات والعوامل الرياضية المتعارف عليها. مثال ذلك

$$Sum = A + B + C$$

وهذه اللغات تعتبر سهلة ومرغوبة من وجهة نظر المبرمجين بالمقارنة بلغات التجميع ولغة الآلة وذلك لسهولة كتابتها وفهمها وحل المشاكل باستخدامها، ومن أمثلة هذه اللغات لغة C، C++، الباسكال Pascal، الفورتران Fortran، البيسك Basic، الأدا ADA، الجافا Java وغيرها.

ومن المعلوم أن عملية تحويل البرنامج من لغة ذات مستوى عال إلى لغة الآلة تستهلك وقتاً ولذلك تم تطوير نسخ من لغات المستوى العالي بحيث تستخدم برنامج مفسر Interpreter والذي يقوم بترجمة وبالرغم من أن البرامج المترجمة الناتجة من عملية الترجمة باستخدام المترجم compiler تكون أسرع في التنفيذ عن البرامج التي تستخدم المفسر (Interpreter) إلا أنه يفضل وجود نسخة من اللغة تعمل باستخدام المفسر وذلك لسهولة التغيير والحذف والإضافة والتصحيح. وبعد الانتهاء من كل التعديلات والوصول إلى نسخة نهائية فإنه يتم استخدام المترجم لترجمة البرنامج وإنتاج نسخة تنفيذية حتى تكون أسرع في التنفيذ بعد ذلك عند تشغيلها على الحاسب.

أهمية مهنة البرمجة

من المعلوم أن الذي يقوم بكتابة البرامج لحل المشكلات الكثيرة والمعقدة هم المبرمجون ولا يمكن الاستغناء عنهم بحال من الأحوال لأن دورهم مهم وحيوي وتكثر الحاجة لهم في شتى المجالات وذلك لعمل الآتي: -

١ - كتابة برامج وبناء الأنظمة المختلفة لحل المشاكل وتبسيط التعامل مع الحاسب.
٢ - المسئولية الكاملة عن إصلاح ما يحدث من أعطال أو حل المشاكل التي تحدث في الأنظمة المختلفة.
٣ - بناء واجهة المستخدم المختلفة في كثير من اللغات والتطبيقات.
٤ - بناء نظم التشغيل المختلفة مثل Unix، Windows وغيرها من النظم. فمثلاً تستخدم لغة C في بناء نظام التشغيل Unix.
٥ - برامج المواجهة المختلفة في الأنظمة المختلطة الرقمية و التماثلية.

صناعة البرمجيات

تعتبر صناعة البرمجيات في عصرنا الحالي من الصناعات المهمة جدا والتي تتطور باستمرار نتيجة التطور الهائل في صناعة الحاسبات الآلية، ولذلك فإن هذه الصناعة تتطلب مبرمجين مهرة ولديهم القدرة على تحليل وحل المشاكل بالإضافة إلى إلمام بكل المستجدات والعلوم والتطوير المتعلق بالحاسب وصناعة الحاسبات و ذلك حتى يستطيعوا مواكبة تطوير البرامج والنظم المختلفة للاستفادة العظمى من التقدم في الحاسبات.

حل المشكلة

Problem Solving

مقدمة

القدرة على حل المشاكل بواسطة البرمجة هي مهارة وطريقة مرتبة ولا تعتمد على العشوائية، وهذه القدرة يمكن اكتسابها وتعلمها باتباع بعض القواعد التي تساعد على ذلك، وبعض هذه القواعد ذكرها رين ديكارت الرياضي والفيلسوف المعروف وهي: -

١ - لا يمكن قبول أي شيء حقيقة مسلمة إلا إذا ثبت ذلك بالتجربة والمشاهدة.

٢ - كل مشكلة أو معضلة يتم تبسيطها وتقسيمها إلى أجزاء عدة كلما أمكن ذلك.

٣ - فكر بطريقة منظمة ومنطقية وذلك بالبدء بالأجزاء البسيطة والسهلة الفهم ثم التدرج إلى الأجزاء الأصعب وهكذا حتى يتم الانتهاء من المشكلة.

٤ - المراجعة لجميع الأجزاء حتى يكتمل الحل.

وبالرغم من أن هذه القواعد تم وضعها قبل ٣٠٠عام من صناعة أول حاسب إلكتروني إلا أنها ما زالت مطبقة وصالحة للاستخدام، والتفكير الجيد والمنظم لتعريف وتحديد المشكلة ضروري ومهم جداً وأساسي للحصول على نتائج صحيحة وبخاصة عند التعامل مع الحاسب، ولذلك فإن أول خطوة لحل المشكلة هو فهمها.

الخوارزميات (Algorithms)

لقد استخدمت كلمة الخوارزمية، في القرن الماضي، وبشكل واسع، في أوروبا وأمريكا، وكانت تعني، الوصف الدقيق لتنفيذ مهمة من المهمات، أو حل مسألة من المسائل. وقد اشتق الغربيون هذه الكلمة من اسم عالم الرياضيات المسلم المعروف، محمد بن موسى الخوارزمي.

وتستخدم كلمة الخوارزمية، على نطاق واسع، في علوم الرياضيات والحاسب، الآن حيث تعرف بأنها:

مجموعة الخطوات (التعليمات) المرتبة، لتنفيذ عملية حسابية، أو منطقية، أو غيرها بشكل تتابعي متسلسل ومنظم.

إن أي خوارزمية تتكون من خطوات مرتبة، بعضها إثر بعض، وكل خطوة تعتبر بنفسها وحدة من وحدات البناء الكامل للخوارزمية، ويختلف حجم هذه الخطوات باختلاف الخوارزميات، واختلاف الأشخاص، الذين يقومون بتنفيذ تلك الخطوات. والمثال التالي يوضح معنى الخوارزمية:

مثال:

إذا أردنا أن نوجد متوسط درجات الحرارة : T_3, T_2, T_1 مثلاً فإن خطوات الحل المنطقية يمكن ترتيبها في الخوارزمية التالية:

الخطوة الأولى: اقرأ قيم درجات الحرارة : T_3, T_2, T_1

الخطوة الثانية: احسب متوسط درجات الحرارة، AV، من المعادلة:

$$AV = (T_1 + T_2 + T_3) / 3$$

الخطوة الثالثة: اطبع النتيجة

خرائط التدفق Flow charts

تستخدم خرائط التدفق في بيان خطوات حل المسألة وكيفية ارتباطها ببعض، باستخدام رموز اصطلاحية لتوضيح خطوات الحل، وهذه الرموز مبينة بشكل رقم (1-2)

أهمية استخدام خرائط التدفق:

من أهم فوائد استخدام خرائط التدفق قبل كتابة أي برنامج، الأمور الآتية:

١. تعطي صورة متكاملة للخطوات المطلوبة لحل المسائل في ذهن المبرمج، بحيث تمكنه من الإحاطة الكاملة بكل أجزاء المسألة من بدايتها وحتى نهايتها.
٢. تساعد المبرمج على تشخيص الأخطاء التي تقع عادة في البرامج، وبخاصة الأخطاء المنطقية منها، والتي يعتمد اكتشافها على وضع التسلسل المنطقي، لخطوات حل المسألة لدى المبرمج.
٣. تيسر للمبرمج أمر إدخال أي تعديلات، في أي جزء من أجزاء المسألة، بسرعة، ودون الحاجة لإعادة دراسة المسألة، برمتها من جديد.
٤. في المسائل التي تكثر فيها الاحتمالات والتفرعات، يصبح أمر متابعة دقائق التسلسل، أمراً شاقاً على المبرمج، إذا لم يستعن بمخطط تظهر فيه خطوات الحل الرئيسة بشكل واضح.

الرمز	الحدث الذي يمثله	مثال
(شكل بيضاوي)	حدث طرفي Terminal لبيان بدء (Start) أو انتهاء (Stop) خريطة سير العمليات	START STOP
(مستطيل)	عملية حسابية (Process)	LET X+Y
(متوازي أضلاع)	إدخال/ إخراج INPUT \ OUTPUT لبيان إدخال/ إخراج معلومة من/ إلى الحاسب	INPUT X, Y PRINT Z
(معين)	اتخاذ قرار Decision	NO X=Y YES
(أسهم)	اتجاه تدفق (سريان) Flow line	
(سداسي)	تكرار أو دوران Loop	FOR I= 1 to 10

من أهم فوائد استخدام خرائط سير العمليات قبل كتابة البرنامج لمسألة ما ما يأتي:

١.تمكن المبرمج من الإلمام الكامل بالمسألة المراد حلها و السيطرة على كـل أجزائها بحيث تساعده على اكتشاف الأخطاء المنطقية (Logic Error) و التي تعتبر من أهم الأخطاء التي تجهد المبرمج.

٢. تساعد بيسر و سهولة على تعديل البرامج الموضوعة بمجرد النظر.
3.يعتبر الاحتفاظ برسوم خرائط سير العمليات لحلول مسائل معينة أمراً مهماً إذ يكون مرجعاً عند إجراء تعديلات عليها أو استخدامها لحل مسائل أخرى مشابهة دون الحاجة إلى الرجوع إلى المبرمج الأول باعتبار أن الحلول الأولى قد صيغت في خطوات واضحة بسيطة و مفهومة.

٤. توفير وسيلة مناسبة ومساعدة في كتابة البرامج ذات التفرعات الكثيرة.

من أنواع الخرائط

* خرائط التتابع البسيط (Simple sequential Flowchart).
* خرائط التفرع (Branched Flowchart).
* خرائط الدوران البسيط (Loop Flowchart).
* خرائط الدورانات المتداخلة (Nested).

أولاً: خرائط التتابع البسيط (Simple sequential Flowchart).

يخلو هذا النوع من التفرعات Branches و الدورانات loops، و يكون الشكل العام لهذا النوع كما هو مبين في الشكل



و كلمة Event الواردة في شكل تعني الحدث أو العملية المطلوب تنفيذها.

مثل : أرسم خريطة سير العمليات لإيجاد مساحة و محيط دائرة نصف قطرها معلوم R.

مساحة الدائرة = πR^2
محيط الدائرة = $2\pi R$
حيث π = النسبة التقريبية
وقيمتها العددية ثابتة و تساوي ٣,١٤ بينما R متغير.
وتكون خطوات الحل المبينة في الشكل كما يلي:

١.ابدأ.

٢.اقرأ قيمةR .

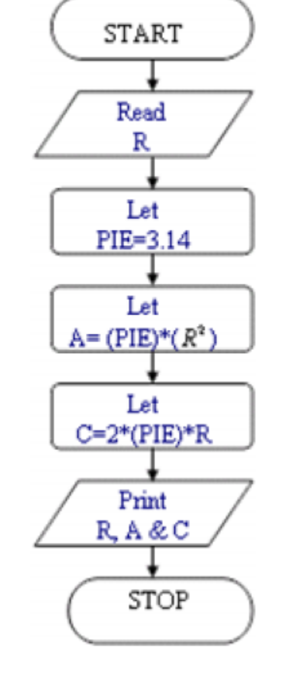
٣. ضع قيمة 3.14=PIE

٤. احسب المساحة(A) من المعادلة A =(PIE)*R*R.

٥. احسب المحيط (C) من المعادلة C =2*(PIE)*R.

٦. أطبع قيم كل من C, A, R.

٧. توقف.



مثل:ارسم خريطة سير العمليات لحساب قيمة كل من المتغيرات C, B, A في المعادلة الاتية:

$A= X^2 + 2y ... (1)$
$B= 2X - 3A ... (2)$
$C= A^2 +XB ... (3)$

إذا علمت أن قيم كل من Y, X معطاة (معلومة)، ثم أطبع قيم كل من C, B, A, Y, X.
الحل: من الواضح أنه يمكننا من حساب قيمة المتغير A في المعادلة(١) لمعرفتنا بقيم المعطيات الأولية X ،Y, ويمكننا من حساب قيمة المتغير B في المعادلة (٢) بالاعتماد على قيمة X المعلومة لدينا وقيمة المتغير Aالمحسوبة في الخطوة السابقة، أما قيمة المتغير C في المعادلة (٣) بالاعتماد على قيم كل من المتغيرات B, A, X وكلها معلومة.

وتكون خطوات حل المسألة :

١.ابدأ.

٢.اقرأ قيمة كل من Y, X.

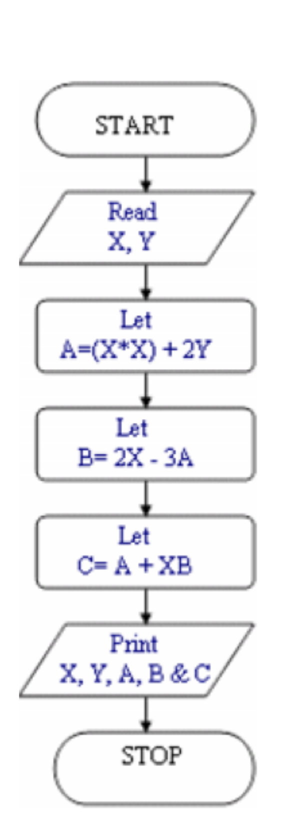
٣.احسب قيمة A من المعادلة (١).

٤.احسب قيمة B من المعادلة (٢).

٥.احسب قيمة C من المعادلة (٣).

٦.اطبع قيمة كل من C, B, A, Y, X.

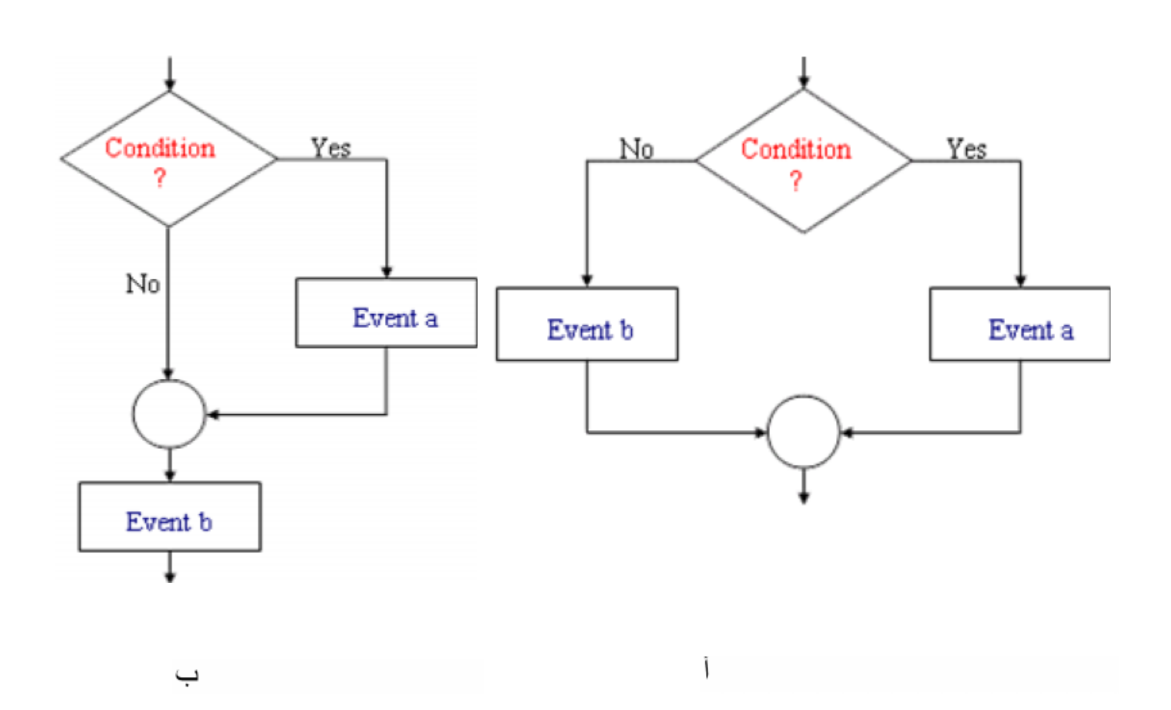
٧.توقف.



• خرائط التفرع (Branched Flowchart):

ويحدث التفرع في البرامج بسبب الحاجة لاتخاذ قرار أو مفاضلة بين اختيارين أو أكثر، وهناك أسلوبان في تنفيذ القرار وبشكل عام فإن خرائط التفرع يمكن أن تأخذ إحدى الصورتين الاتيتين :

يمكننا ملاحظة أن شكل "أ" يبين أنه إذا كان جواب الشرط YSE (Condition) فإن الحــدث التالي في التنفيذ يكون الحدث (a) أما إذا كان الجــواب NO فــإن الحــدث التــالي يكــون الحدث(b) كما يمكننا أن نلاحظ في الشكل "ب" أنه إذا كان جواب الشرط YSE فإن الحــدث التالي في التنفيذ يكون الحدث (a) ثم يتبعه الحدث (b) أما إذا كان جواب الشرط NO فــإن الحدث التالي يكون الحدث (b) مباشرة.



ب أ

مثل:ارسم خريطة سير العمليات لإيجاد قيمة الاقتران F(x) المعرف حسب القاعدة التالية:

F(X) = |X|	X if X>=0
	-X if X<0

حيث كلمة (if) هنا تعني عندما.

١. ابدأ

٢. اقرأ قيمة المتغير X .

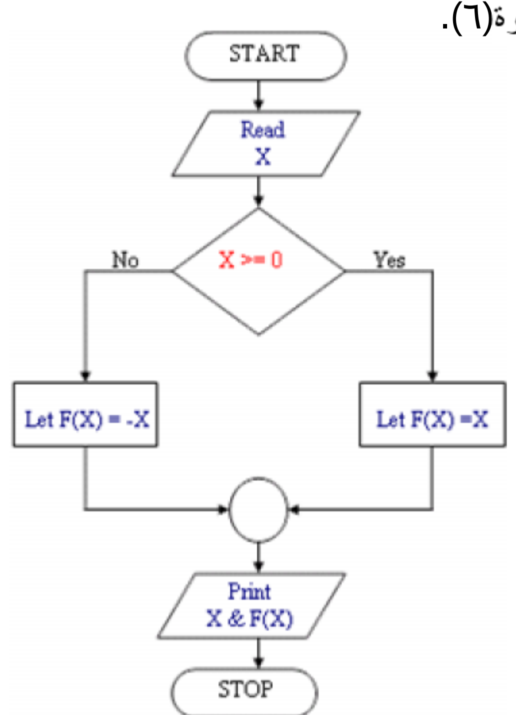
٣. إذا كانت X أكبر أو تساوي صفرًا اذهب إلى خطوة(٤) وإلا فاذهب إلى الخطوة(٥).

٤. احسب قيمة الاقتران من F(X)=X ثم اذهب إلى الخطوة(٦).

٥. احسب قيمة الاقتران من F(x)= -X.

٦. اطبع قيمة كل من X ،F(x).

٧. توقف.

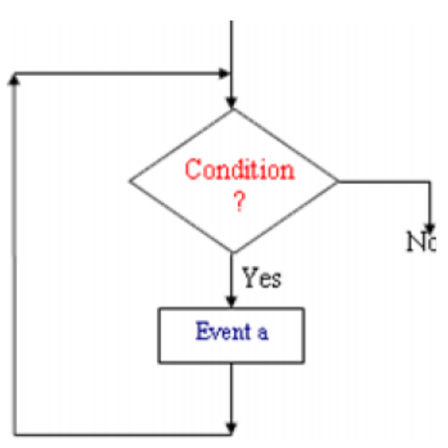


• خرائط الدوران البسيط (Loop Flowchart) :

وهذه الخرائط نحتاج إليها لإعادة عملية أو مجموعة من العمليات في البرنامج عددًا محدودًا أو غير محدود من المرات، ويكون الشكل العام لمثل هذه الخرائط كما يلي .

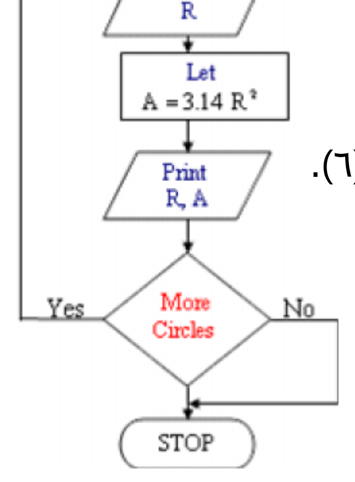
Event a
Condition ?
No
Yes

الحدث (a) يتكرر تنفيذه في كل دورة حتى يصبح جواب الشرط YES.



الحدث (a) يتكرر تنفيذه في كل دورة طالما كان جواب الشرط YES.

مثل: ارسم خريطة سير العمليات لإيجاد مساحة مجموعة من الدوائر أنصاف أقطارها معلومة:



١. ابدأ.
٢. اقرأ نصف قطر الدائرة (R).
٣. أوجد مساحة الدائرة (A).
٤. اطبع قيم كل من A, R.
٥. هل هناك مزيد من الدوائر؟
فإن كان نعم فعد إلى الخطوة(٢) وإن كان لا فعد إلى الخطوة (٦).
٦. توقف.

العداد Counter:

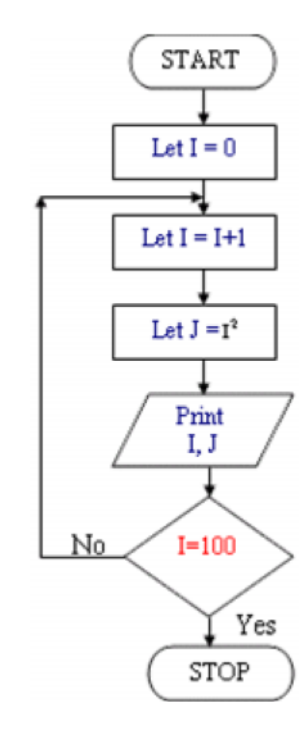
في كثير من الأحيان نحتاج في برامج الحاسب الالكتروني إلى العد Counting، فقد نريد مثلاً أن نعد عدد كل من الطلاب والطالبات ضمن الشعبة، وقد تكون هذه العملية سهلة للإنسان لأنها أصبحت ضمن قدراته العقلية التي يكتسبها من الطفولة، إلا أن الحاسب يحتاج إلى تصميم خوارزمية للعد Counting Algorithm تتضمن خطوات معينة إذا اتبعتها استطاع أن يعد.

ويمكن تحديد الخطوات التي يتبعها الحاسب حتى يتمكن من العد في الخطوات الأساسية:
١.اجعل العداد مساويًا للصفر.

٢.اجعل القيمة الجديدة للعداد تساوي القيمة القديمة لها زائد واحد، أي أن:
قيمة العداد (الجديدة)= قيمة العداد (القديمة)+١
٣.كرر الخطوات ابتداء من الخطوة ٢.

مثال: ارسم خريطة سير العمليات التي يتبعها الحاسب لطباعة الأعداد الطبيعية من ١ إلى ١٠٠ ومربعاتها.
الحل: خطوات الحل :

١. ابدأ.
٢. اجعل I=0.
٣. اجعل I=I+1.
٤. اجعل $J = I^2$.
٥. اطبع J, I.
٦. إذا كانت I=100 اذهب إلى الخطوة ٧ وإلا اذهب إلى الخطوة ٣.
٧. توقف.



المجاميع الإجمالية:

في كثير من الأحيان نحتاج في برامج الحاسب الإلكتروني إلى جمع مجموعة كبيرة من الأعداد التي تمثل معطيات ظاهرة معينة، فمثلاً قد نرغب في إيجاد الوسط الحسابي لأعمار طلاب الجامعة، ولتحقيق هذا أولاً يجب أن نحسب مجموع أعمار الطلاب، وطبعًا ليس عمليًا إعطاء رمز أبجدي لكل عمر طالب فقد تحتاج لأكثر من عشرة الالاف رمز، في مثل هذه الحالات نصمم خوارزمية معينة للتجميع تسمى خوارزمية التجميع summers Algorithm تتضمن خطوات محددة إذا اتبعها الحاسب استطاع أن يجمع أي كمية من البيانات باستخدام متغيرين اثنين إحداهما هو المتغير الذي نجمعه والآخر هو الجمع الإجمالي (المجمع)، ويمكن تحديد الخطوات التي يجب أن يتبعها الحاسب لتحقيق ذلك في أربع خطوات هي:

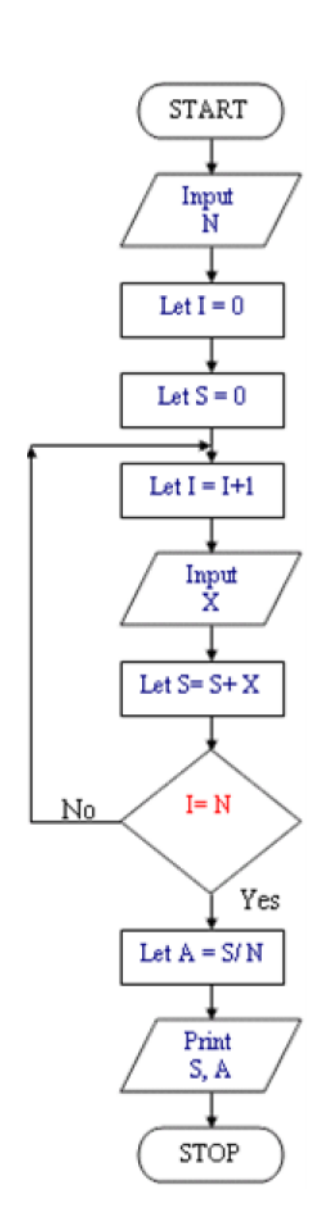
١. اجعل المجمع مساويًا الصفر.
٢. ادخل قيمة واحدة للمتغير.
٣. اجعل القيمة الجديدة للمجمع تساوي القيمة القديمة له زائد القيمة المدخلة للمتغير، أي أن: قيمة المجمع الجديدة=قيمة المجمع القديمة + آخر قيمة مدخلة للمتغير.
٤. كرر ابتداءًا من الخطوة الثانية.

مثل:ارسم خريطة سير العمليات لإيجاد الوسط الحسابي لأعمار طلاب شعبتك.

الحل: نفترض أن إجمالي عدد الطلاب =N ونستخدم عددًا لرقم كل طالب ونرمز له بالرمز I ونرمز لعمر الطالب بـX ونستخدم مجمعًا لأعمار الطلبة ونرمز له بالرمز S ونستخدم الرمز A ليدل على معدل أعمار الطلبة.
وتكون خطوات الحل :

ابدأ.

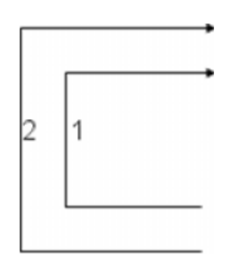
١. ادخل إجمالي عدد الطلاب (N).
٢. اجعل I=0.
٣. اجعل S=0.
٤. اجعل I=I+1.
٥. ادخل X.
٦. اجعل S=S+X.
٧. إذا كانت I=N اذهب إلى الخطوة ٩ وإلا اذهب إلى الخطوة ٥.
٨. اجعل A=S/N.
٩. توقف.





خرائط الدورانات المتداخلة (Nested):

في هذه الحالة تكون الدورانات داخل بعضها البعض بحيث لا تتقاطع فإذا كان لدينا مثلاً دورانات من هذا النوع فيسمى الدوران قم (١) دورانًا داخليًا (Inner Loop) بينما الدوران رقم (٢) دورانًا خارجيًا (Outer Loop) ويتم التناسق في عملي في مثل هذين الدورانين بحيث: تكون أولوية التنفيذ للدوران الداخلي.



١. ابدأ.
٢. خذ قطعة.
٣. اقطع منها قطعة طولها ٣ متر.
٤. هل المتبقي يزيد عن ٣ متر؟
إذا كان الجواب نعم فاذهب إلى الخطوة(٣). وإذا كان الجواب لا فاذهب إلى الخطوة (٥).
٥. هل هناك مزيد من القطع المراد تقطيعها ؟ إن كان الجواب نعم فاذهب إلى الخطوة(٢) وإن كان لا فاذهب إلى الخطوة(٦).
٦. توقف.



مقـــدمة



فيجوال بيسك ٢٠٠٨ هو أداة تطويرية تستخدم لإنتاج التطبيقات والبرامج ويحتوي على العديد من الإعدادات و الأكواد الجاهزة التي تيسر علينا كتابة الكود وتصميم البرامج التي تعمل تحت منصة الويندوز ، وكذلك البرامج التي تعمل على الانترنت والأجهزة الكيفية (أجهزة الجيب).
فيجوال بيسك يساعد على زيادة الإنتاجية عند تصميم البرامج خاصة البرامج المتعلقة بقواعد البيانات وبرامج الانترنت.

بدأت الفيجوال دوت نت في فبراير ٢٠٠٢ ثم نسخة فيجوال دوت نت ٢٠٠٣ في مارس ٢٠٠٣ ثم الفيجوال دوت نت ٢٠٠٥ في أواخر ٢٠٠٥ . وبعد العديد من التطويرات التي تمت على لغة الفيجوال دوت نت ٢٠٠٥ تم إصدار الفيجوال دوت نت ٢٠٠٨ في بداية عام ٢٠٠٨. ويعتبر فيجوال بيسك دوت نت ٢٠٠٨ مربوطاً ربطاً وثيقاً مع الفيجوال المرئي ٢٠٠٨ والذي يشمل على السي شارب ٢٠٠٨ والسي بلس ٢٠٠٨ والعديد من لغات التطوير.

تمهيد

تتكون .Net من الطبقات تجمع في ملفات تسمى Assemblies ومن المهم عدم الخلط بين الملفات Assemblies وبين الفئات فالملف Assemblies هو الملف أو الملفات الفعلية التي تحتوي على الكود للطبقات الفئات على سبيل المثال كل الطبقات في الفئة System.IO تقع في الملف Mscorlib.dll وهو الملف الفعلي الموجود على قرص التخزين .

Assembly Name	Assembly File	Purpose
mscorlib	mscorlib.dll	Core system types
System	System.dll	CLR-specific system types
System.Data	System.Data.dll	ADO.NET
System.DirectoryServices	System.DirectoryServices.dll	Active Directory
System.Drawing	System.Drawing.dll	Windows graphics functionality
System.EnterpriseServices	System.EnterpriseServices.dll	Services formerly known as COM+ 1.0
System.Management	System.Management.dll	Windows computer management
System.Messaging	System.Messaging.dll	MSMQ messaging services
System.Security	System.Security.dll	Programmatic security
System.Web	System.Web.dll	ASP.NET
System.Web.Services	System.Web.Services.dll	Additional Web service support for ASP.NET
System.Windows.Forms	System.Windows.Forms.dll	Windows Forms framework
System.XML	System.XML.dll	Support for programming in Extensible Markup Language (XML)
Microsoft.VisualBasic	Microsoft.VisualBasic.dll	Visual Basic methods, constants, and attributes

بيئة فجول بيسك دوت نت 2008

هي احد اللغات المتاحة داخل برنامج VISUAL STUDIO.NET بالإضافة إلى

- VISUAL C#.NET
- VISUAL J#.NET
- VISUAL C++.NET

التعرف على نافذة البرنامج

بعد تحميل البرنامج ، من قائمة ابدأ > كافة البرامج > Microsoft Studio 2008 > اختر Microsoft Studio 2008 إذا كانت هذه هي المرة الأولى لكي في فتح البرنامج سوف يتأخر البرنامج لبضع لحظات لتركيب الإعدادات الخاصة، سيطلب منك خلال الفتح اختيار اللغة التي تعمل عليها (فيجوال بيسك أو سي شارب) اختر فيجوال بيسك ليختار لكي إعدادات بيئة الفيجوال بيسك لتظهر لكي أولاً START PAGE و بها تظهر آخر تطبيقات تم إنشائها أو تعديلها.

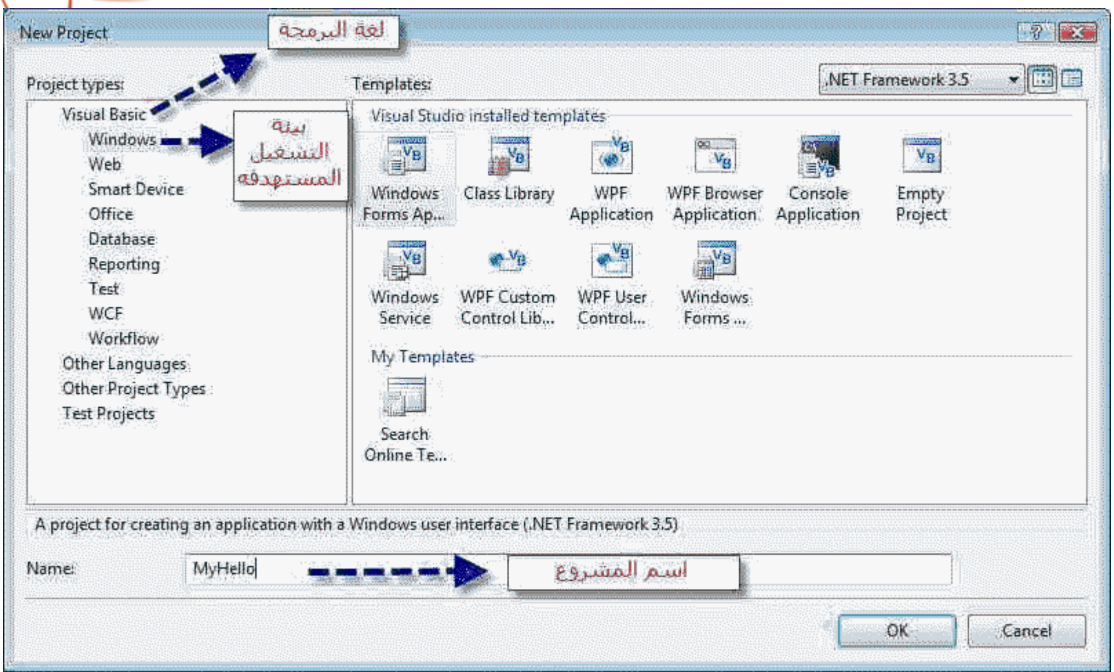
صورة الصفحة الأولى (١)

إنشاء مشروع جديد

الآن قوم بالضغط على قائمة FILE و اختار منها New Project ستظهر لك نافذة New Project لتختار منها التطبيق الذي تريدي إنشائه.

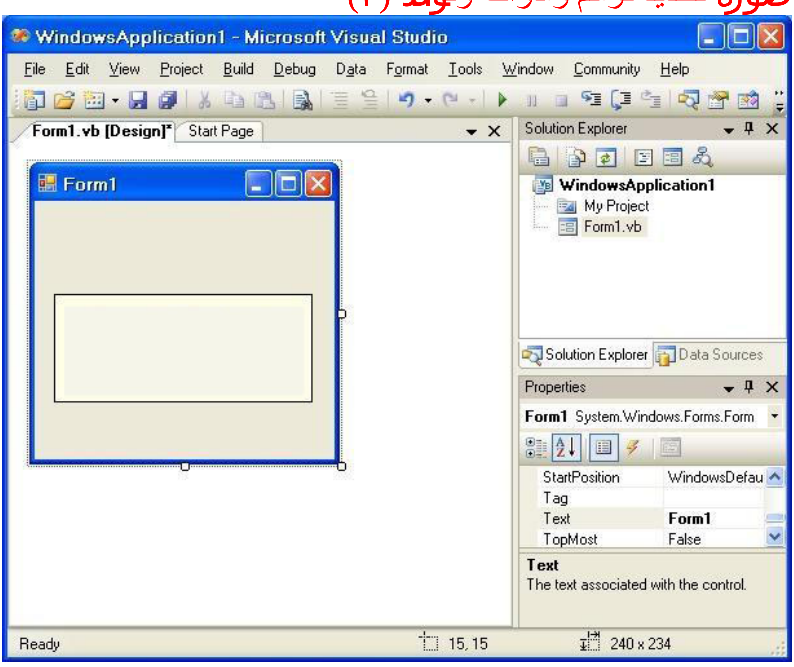
صورة تحديد التطبيق المطلوب (٢)

خطأ!



على اليسار يوجد نوع المشروع الذي تريد إنشاء احد تطبيقاته وعلى اليمين تظهر تطبيقات المشروع الذي حددناه وفي أسفل النافذة اسم المشروع و يمكن أن نغيره إلى الاسم اللذي نريده. نختار الآن احد تطبيقات VISUAL BASIC.NET ونختار من هذه النافذة WINDOWS APPLICATION ثم نضغط موافق سوف تظهر لنا قوائم وأدوات ونوافذ لغة VISUAL BASIC.NET

صورة تحديد قوائم وأدوات ونوافذ (٣)



إظهار القوائم ونافذة الأدوات

- لإظهار قوائم الخصائص أو الأدوات نقوم بوضع المؤشر على أسماء القوائم الموجودة.
- لإظهار نافذة الأدوات من القائمة View واختر Toolbox أو بالضغط على Ctrl+Alt+X

-إذا ظهرت النافذة بشكل منطوي تحت احد جوانب نافذة الفيجول استديو ما علينا إلا الضغط على الزر الأوسط الموجود في أعلى أي نافذة.

صورة الزر (٤)

والآن ماذا لاحظنا بالصورة نافذة المشروع الأساسية بالوسط ظاهراً بها فورم اسمه التلقائي Form1 ، في الجهة اليسرى تظهر قائمة الأدوات ,ويظهر بالجهة اليمنى نافذة Solution Explorer وأسفل منها نافذة الخصائص Properties .

ولكن ما هو الفرق بين Solution (المشروع) ول Project (التطبيق) ؟

١- Project أو التطبيق:

هو عبارة عن محتوى للمكونات الخاصة بتطبيق ما مثل النماذج Forms وكل الأكواد البرمجية كل Modules أو Classes .

٢- Solution أو المشروع :

مع بداية ظهور لغة Vb.Net ظهر ما يسمي Solution المشروع وفائدته تنظيمية لل Projects ، ويعتبر أل Solution مساحة العمل الخاصة بالدوت نت فمن الممكن أن يحتوي على أكثر من تطبيق Project بداخلة ، وتستطيع أيضا إضافة مكونات خارجة عن أطار الكود لل Solution أي حتى ولو لم تتعلق بكود التطبيق كملف وورد أو صورة أو ملف صوتي وبالتالي تستطيع استخدامها داخل تطبيق الدوت نت.

نافذة Solution Explorer

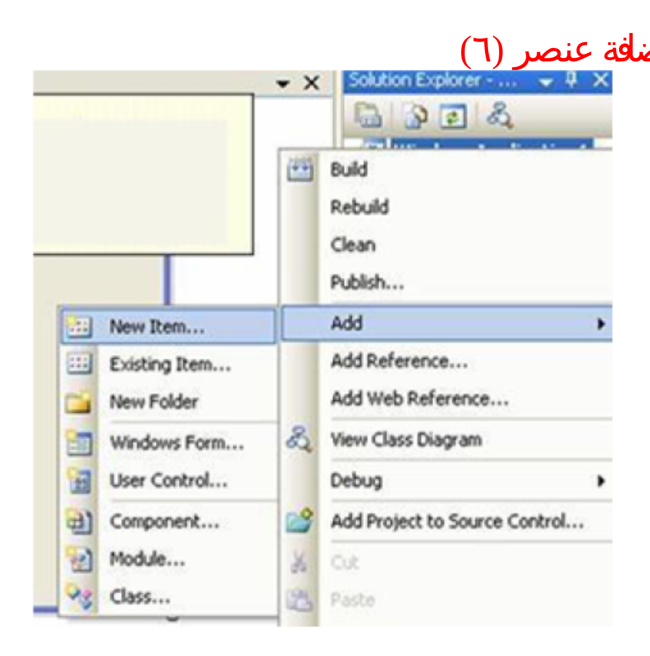
- توجد النافذة Solution Explorer في الجهة اليمني كما يمكن إظهارها من قائمة View أو بالضغط على Ctrl+Alt+L وتعمل على عرض عناصر ومكونات التطبيق على هيئة قائمة شجرية في اعلي القائمة يظهر اسم التطبيق الحالي ويليه بقية مكونات التطبيق

صورة لنافذة Solution Explorer (5)



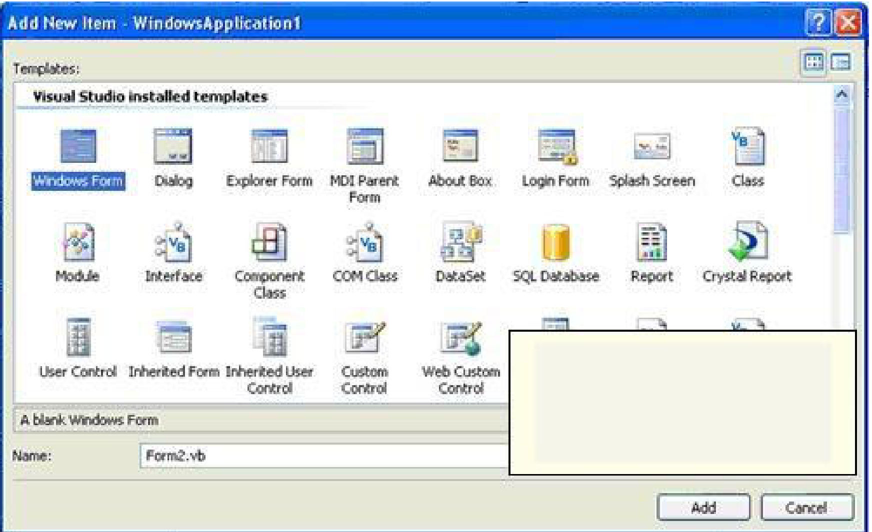
ونستطيع إضافة عناصر إلى التطبيق الحالي من خلال النافذة Solution Explorer وذلك بالضغط على ملف المشروع بزر الماوس الأيمن ثم Add ومنه نختار New Item وبعد الضغط على New Item سوف تظهر لنا نافذة Add New Item

صورة لنافذة إضافة عنصر (٦)



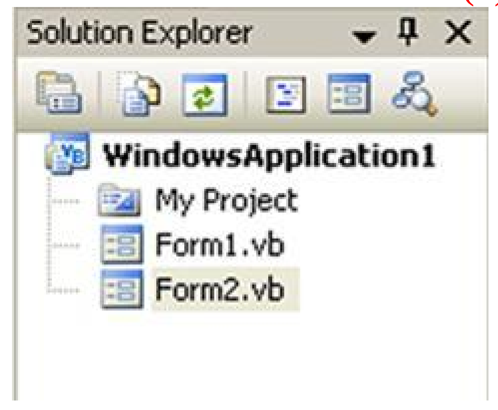
وبها العديد من تطبيقات VISUAL STDIO.NET نختار منها ما نريد إضافته إلى تطبيقنا الحالي Windows Form وهنا سوف نختار مثلا

صورة لنافذة تحديد العنصر المطلوب إضافته (٧)

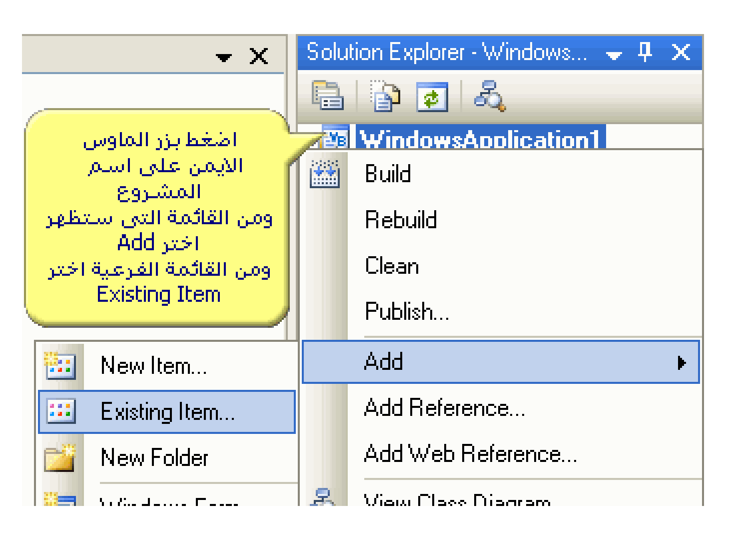


قم بالتأشير علية ويمكننا تغيير اسم التطبيق قبل إضافته من أسفل نافذة Add New Item أو نتركه كما هو ثم نضغط على الزر Add فنلاحظ أضافه Form أخرى إلى التطبيق تأخذ الاسم Form2 ويظهر في قائمة Solution Explorer

صورة للعنصر بعد إضافته (٧)



صورة لإضافة عناصر آخرى مثل صورة وغيرها (٨)

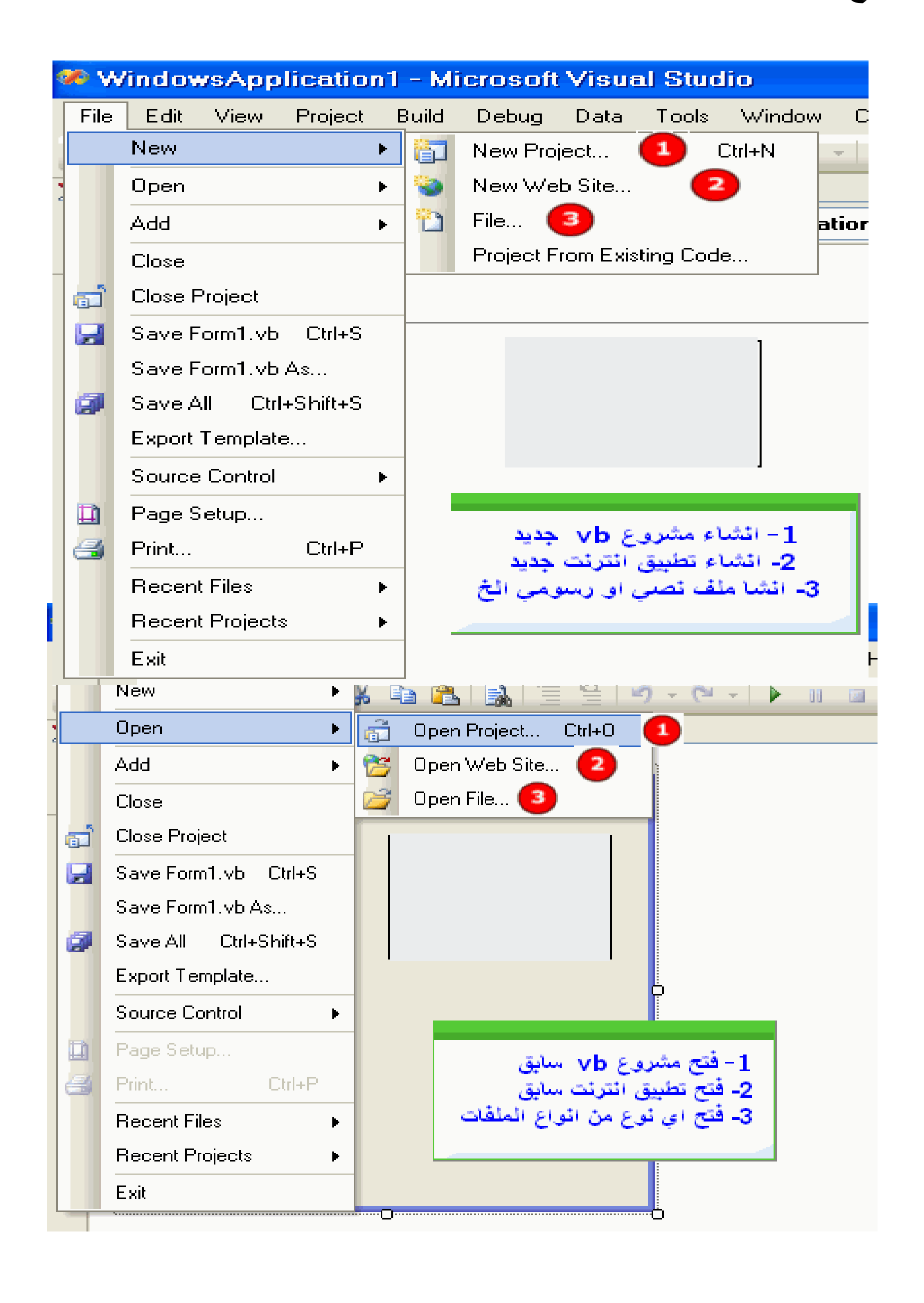




شرح القوائم

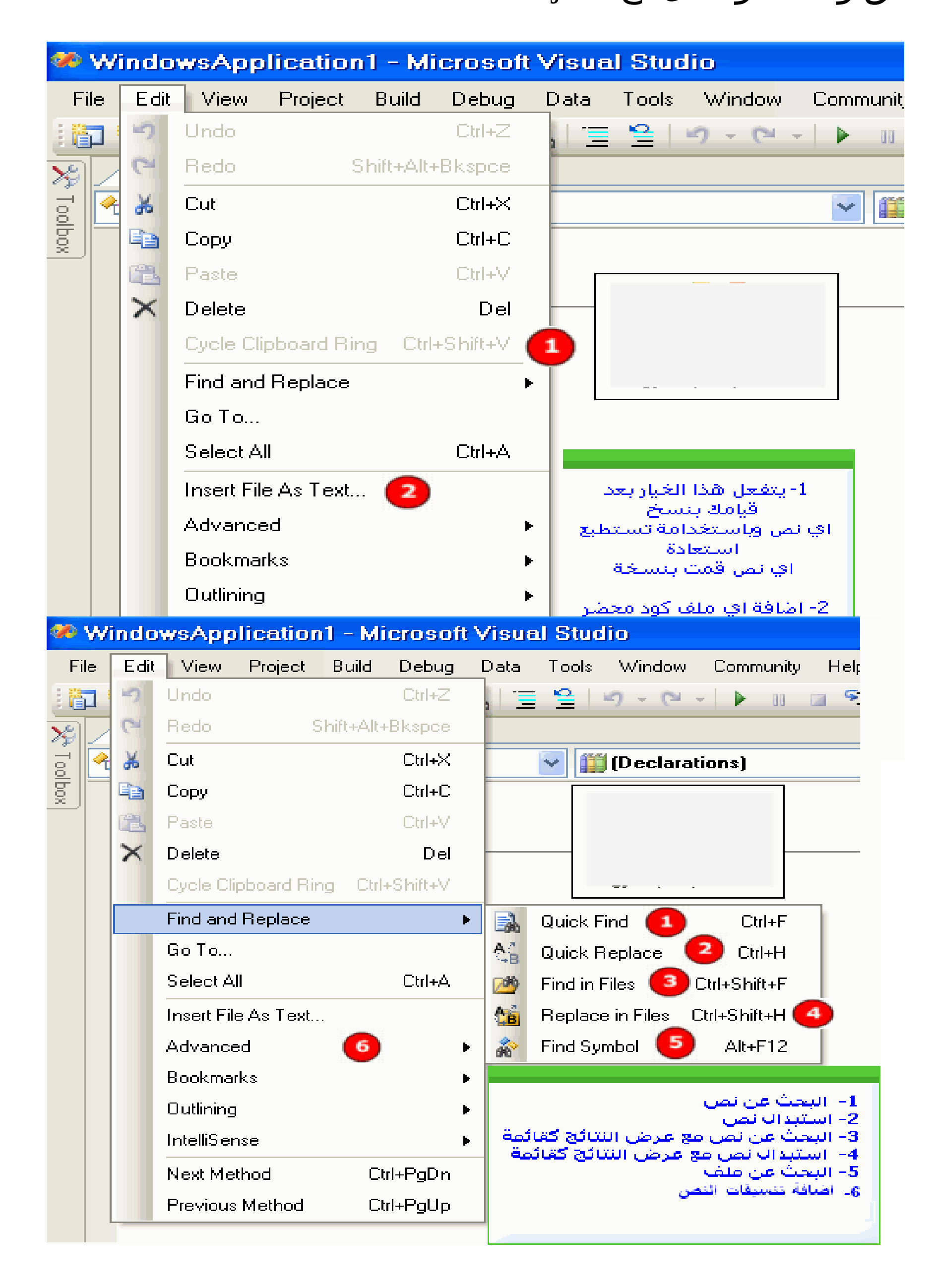
١- قائمة [File]:

وتتكون من أزرار الأوامر التي تتعامل مع إنشاء وفتح وحفظ وإغلاق كلا من الملفات والمشاريع



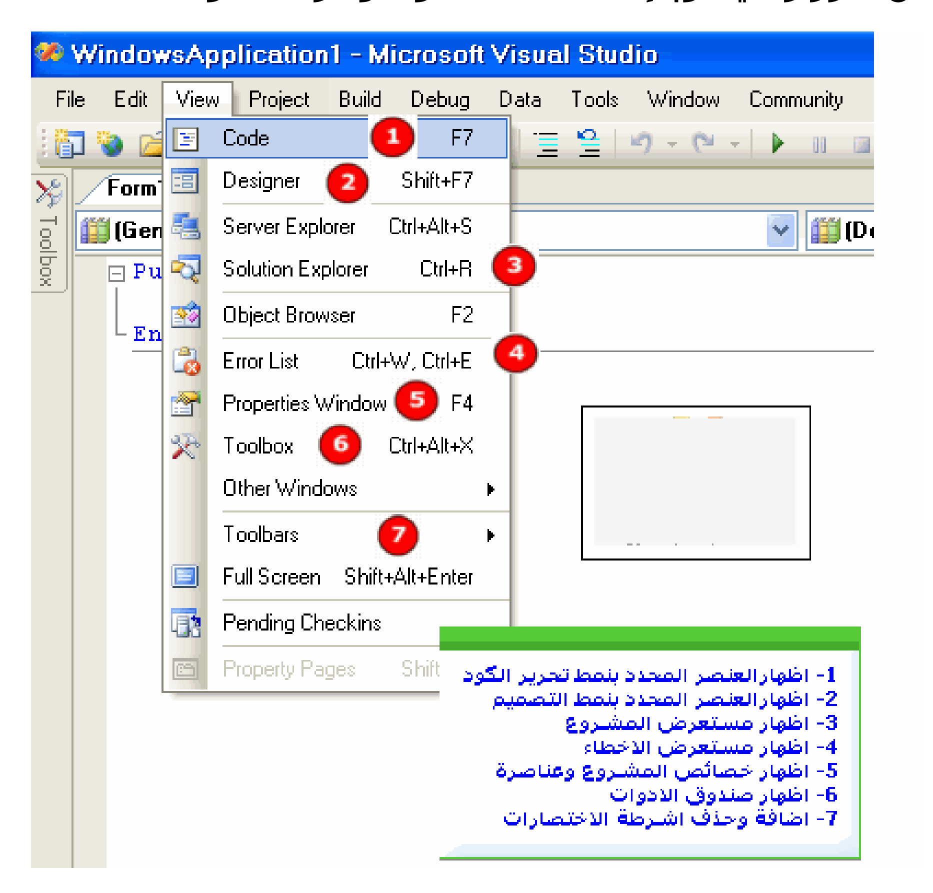
٢ قائمة[Edit]

تحتوي هذه القائمة على الأزرار القياسية التي تعمل على الإلغاء او التراجع والنسخ والقص واللصق والحذف وتتعامل مع text and other objects



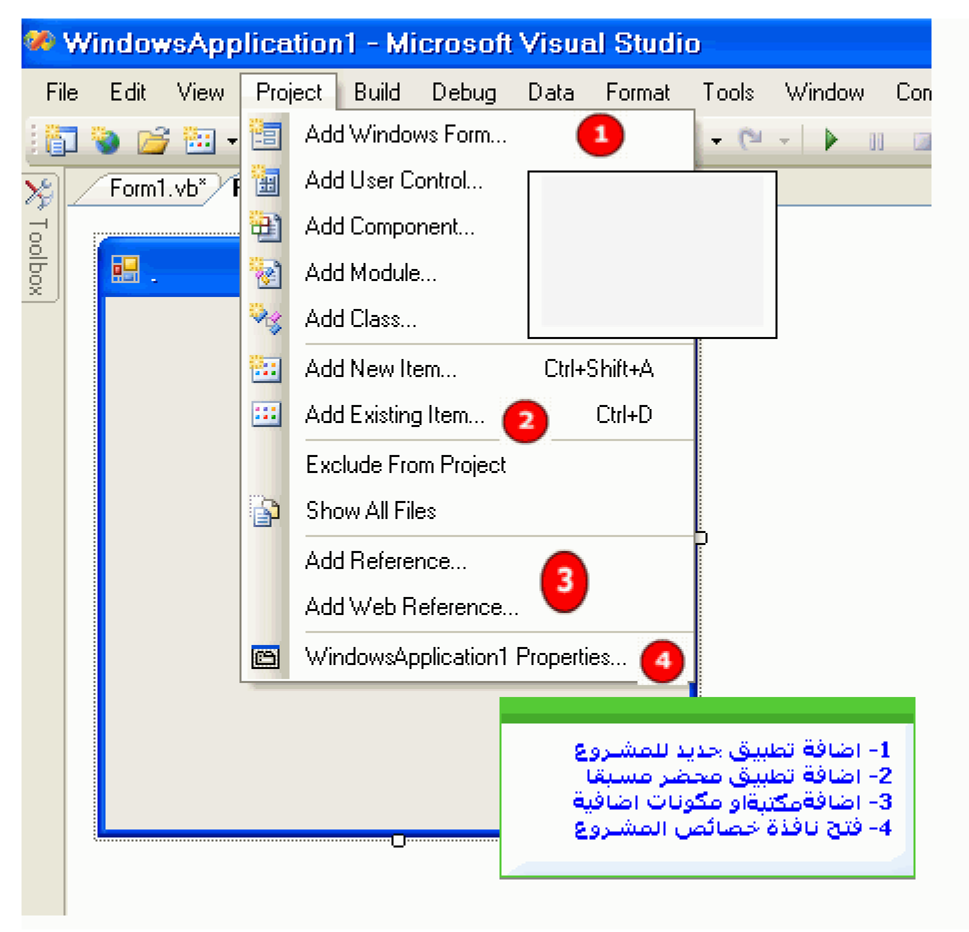
٣ قائمة[View]:

تتكون من الأزرار التي تقوم بإخفاء وإظهار كل نوافذ وأشرطة المحول بيسك دوت نت



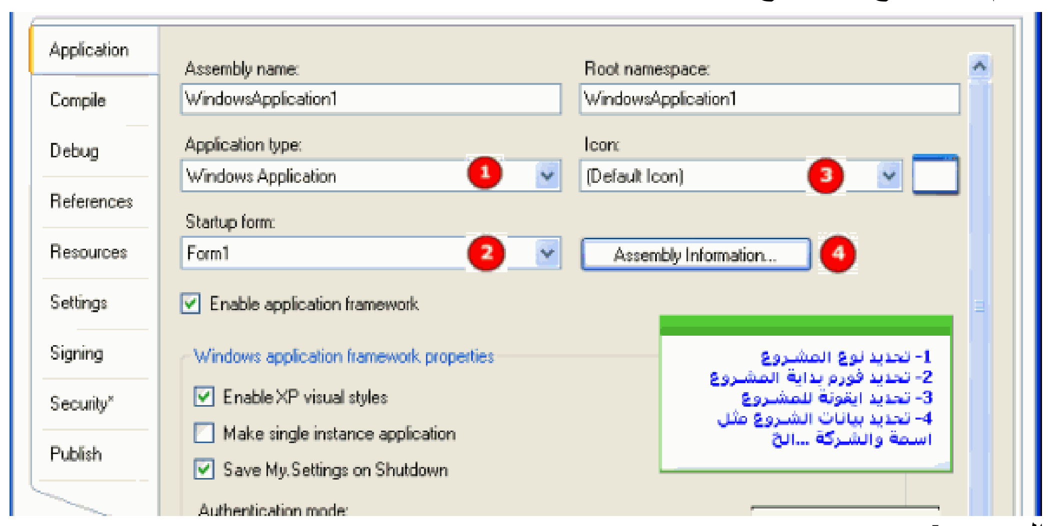
٤ قائمة[Project]:

تتكون من الأزرار التي تقوم بإضافة وحذف العناصر من والى المشروع وهذه القائمة نستطيع القيام ببعض أوامر أزرارها من النافذة Solutions Explorer

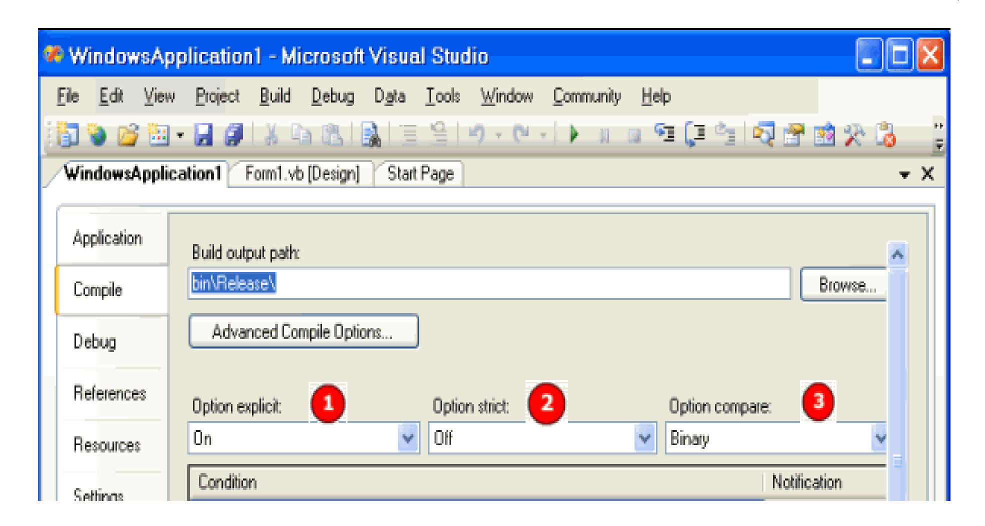


والخيار الأخير بالنافذة كما عرفنا هو لعرض نافذة خصائص المشروع والآن سوف نستعرض أهم خصائص المشرع

أ‌ التبويب Application :
ومنة يتم تحدد نوع المشروع وبياناته



ب التبويب Compile :
ومنة يتم تحديد شكل المعالجة



Option Explicit
إذا كان هذا الخيار مفعل فإنه يلزمك بالإعلان عن كل المتغيرات بالمشروع ويفضل ذلك.
Option strict
إذا كان هذا الخيار مفعل فإنه يلزمك إعطاء أي نوع محدد من البيانات بيانات من نفس النوع
ولو كان الخيار غير مفعل فإنه يقوم بتحويل البيانات المسندة إلى النوع المراد إذا كانت منلسبة .

ويفضل جعل هذا الخيار مفعل لجعل المعالجة أسرع وتحويل أنواع البيانات باستخدام الدوال
أفضل و لجعل هذه الخيارات مفعل لكل مشاريعك اذهب إلى القائمة Tools ومنها اختر
option وقم باختيار Project and solution من قائمة العرض الشجري ومنة

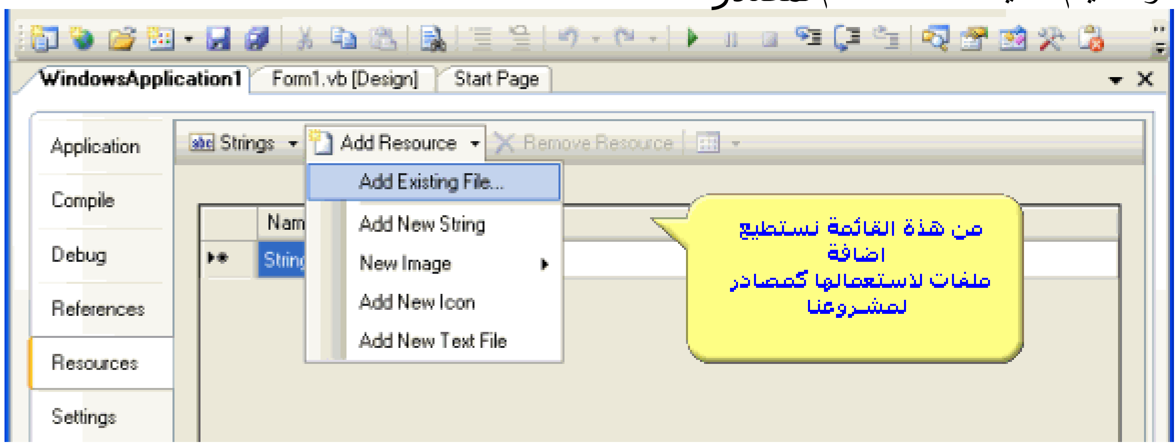
اختر
VB Defaultsوقم بتفعيل الخياران وثم اضغطok .

Option compare

هذا الخيار يحدد با Binary or Text وذلك لتحديد نمط معالجة النصوص اعتماد على البنري أو على تحسس حالة الأحرف ويفضل جعل خيار المعالجة بواسطة Binary لأنه أسرع

ج التبويبResources

ومنة يتم تحديد ملفات تستخدم كمصادر



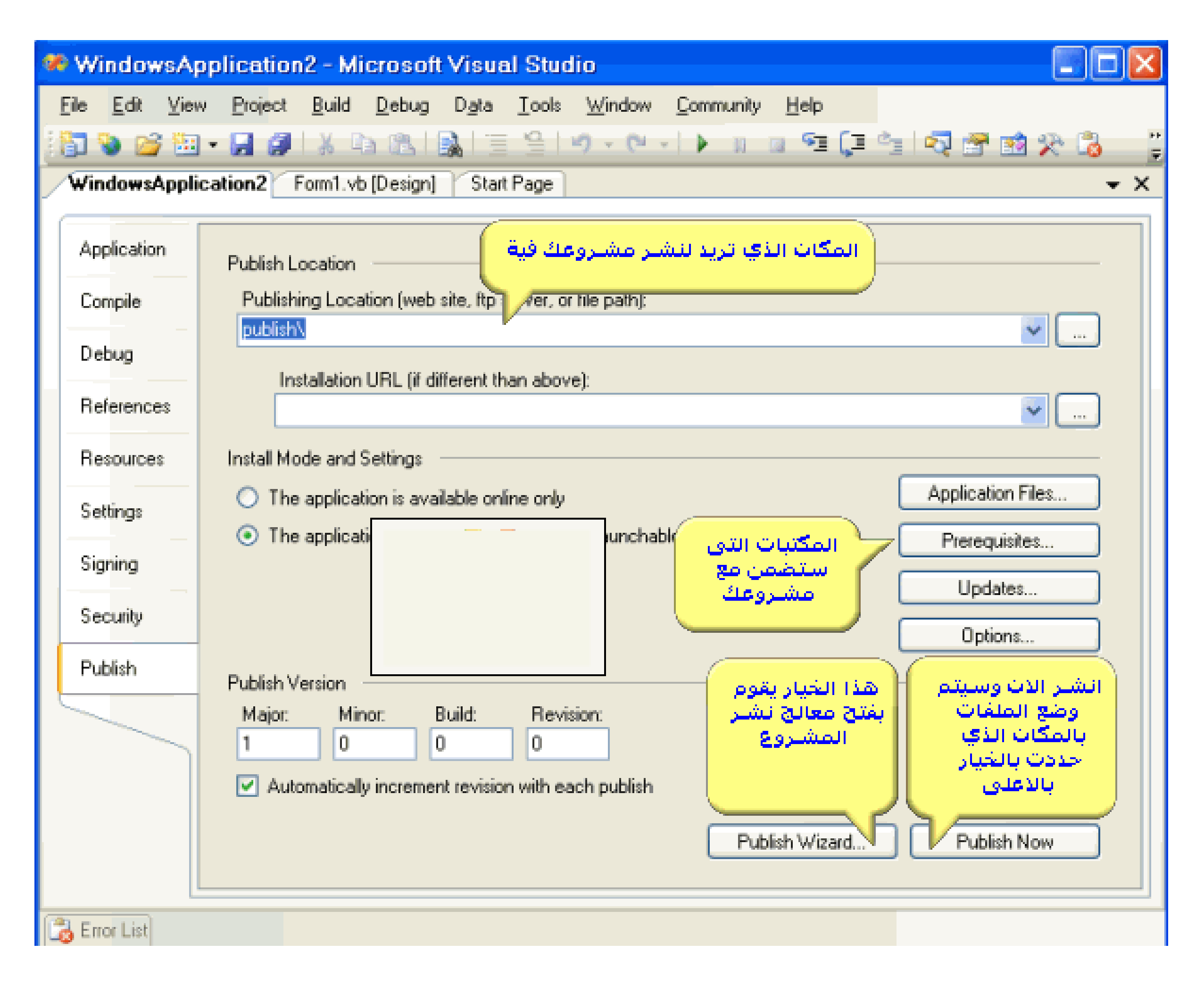
ومن ثم تستطيع التعامل مع الملفات التي أضفتها إلى مشروعك كمصادر بكتابة الكود التالي:

My.Resources.اسم الريسورس الذي تريد

مثلا لو كان لدينا Resources عبارة عن صورة ولتم الصورةCar وأردنا عرض الصورة في Picture Boxما علينا سوى كتابة الآتي

PictureBox.Image = My.Resources.Car

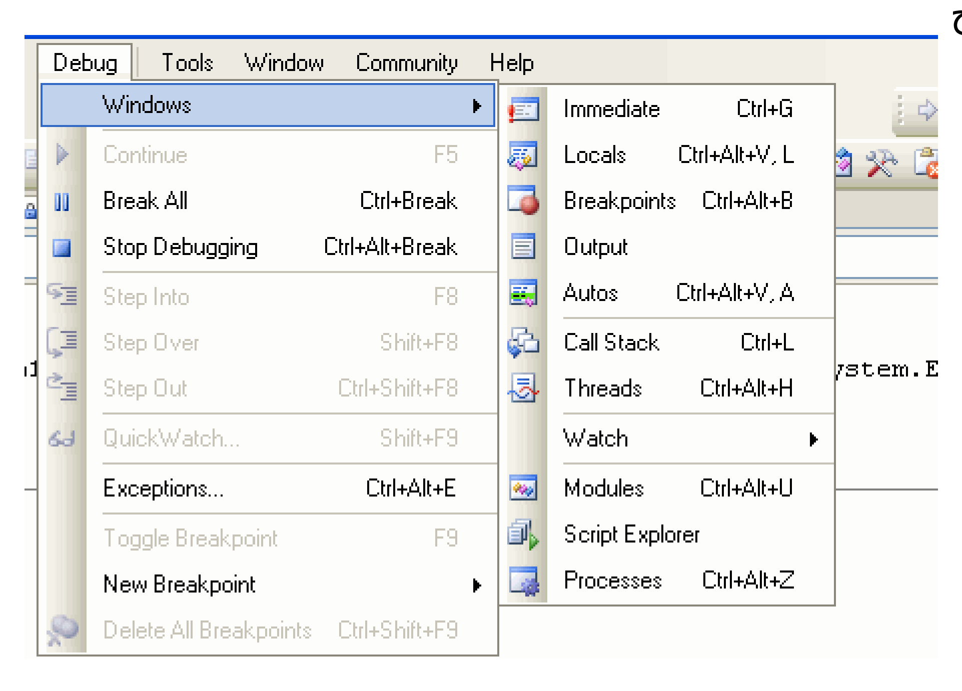
د- التبويب Publish
ومنة يتم نشر المشروع أي تحويله لملف Setup



٥ قائمة [Build]
تتكون من الأزرار التي تعمل على معالجة كلا من التطبيقات والمشاريع



٦- قائمة [Debug]
تحتوي هذه القائمة على أزرار الأوامر التي تقوم بعملية تنقيح وتصحيح الأخطاء البرمجية بكود البرامج



-
-

-

- Window
يحتوي على قائمة فرعية بها الكثير من أدوات تتبع وتصحيح الأخطاء

- Continue
استئناف عملية التنقيح

- Stop Debugging
إيقاف عملية التنقيح

- Detach all
فصل المنقح عن العمليات المرتبطة بيه ولكن لا يقوم بإنهاء تلك العمليات

- Terminate all
إنها أي عملية مرتبطة بالمنقح

- Restart
إيقاف عملية التنقيح الحالية وإعادة تشغيلها من البداية

- Exceptions
يقوم بفتح نافذة الاستثناءات حيث يمكننك تحيد نوع الخطاء الذي تريد أن يتعرف المنقح علية

- New Breakpoint
لإضافة نقطة توقف جديدة

٧- قائمة [Data]

Data Format Tools Window Community Help
Show Data Sources Shift+Alt+D 1
Preview Data... 2
Add New Data Source... 3
Add Query...
Data Sources

1- لعرض نافذة مصدر البيانات التي من خلالها يمكنك اضافة قاعدة بيانات و امكانية عمل الربط الحقول الجداول
2- استعراض بيانات حقول و جداول قاعدة البيانات
3- عرض معالج اضافة قاعدة بيانات

project currently has no da

هذه القائمة تستطيع من خلالها التعامل مع قواعد البيانات من طلقة و عن وطل قواعد البيانات المختلفة Access ,Sql,Oracle

٨- قائمة [Format]
تحتوي على أزرار الأوامر التي تقوم بترتيب وتنظيم الأدوات على الفورم

٩- قائمة [Tools]

تحتوي هذه القائمة على الأوامر التي لا يتلاءم وضعها في أي قائمة أخرى وهي التي تستطيع تغيير بيئة تطوير الفجول نفسها

- Attach To Process

يقوم بفتح نافذة ربط المنقح مع عملية تفاعلية اختر أي عملية ثم اضغط Attach

- Connect to Device

يتيح لك هذا الخيار الاتصال مع أداة أو جهاز مثل الهواتف الذكية

- Connect to Database

يقوم هذا الخيار بفتح نافذة الاتصال مع قاعدة البيانات ومعرفة القواعد المرتبطة مع مقاطع السيرفر الخاص بربط البيانات

- Code Snippets Manager

يقوم بفتح نافذة مدير مقاطع الأكواد لإضافة أو إزالة المقاطع وكيفية استخدام هذه المقاطع من الأكواد الجاهزة يمكنك الضغط بالزر الأيمن في شاشة الكود ثم اختيار Insert Snippet ستعرض لك الأكواد مقسمة بمجلدات حسب الفئة قم باختيار أي مجلد تريد ثم ستعرض أسماء الأكواد الموجودة بالمجلد قم بالضغط على اسم الكود الذي تريد وسيقوم الفجول بيسك بإضافته مباشرتا

- Choose Toolbox Items

يقوم هذا الخيار بفتح نافذة اختيار وحذف الأدوات من Toolbox فالأدوات المضافة تلقائيا بعد تثبيت الفجول ستديو دوت نت ليست كل الأدوات المتاحة استخدامها بل يوجد هناك الكثير من الأدوات التي يمكن إضافتها

- Macros

يحتوي على قائمة فرعية تستطيع من خلالها إنشاء و تعديل أو تنفيذ ميكروا

- External Tools

يقوم بإضافة أو حذف زر من Tools Menu مثلا تستطيع أن تضيف زر يشغل أي برنامج من داخل الفجول بيسك

- Device Emulation Manager

تقوم بفتح نافذة إدارة الأجهزة، تشغيلها أو إغلاقها والاتصال بها

- Import/Export Settings

يقوم هذا الخيار بفتح معالج حفظ أو استعادة إعدادات بيئة التطوير IDE

- Customize

يقوم هذا الزر بفتح نافذة Customize التي تستطيع من خلالها تعديل أزرار القوائم من إضافة وحذف وتعديل مظهر القوائم و إضافة القوائم الفرعية ويمكنك إضافة الأزرار بواسطة السحب والإفلات في أي قائمة تريد

- Options

يقوم بفتح نافذة تستطيع من خلالها تحديد الخيارات الخاصة ببيئة التطوير

١٠- قائمة [Window]
تحتوي هذه القائمة على الأوامر التي تتحكم في نوافذ الفجول بسيك دوت نت

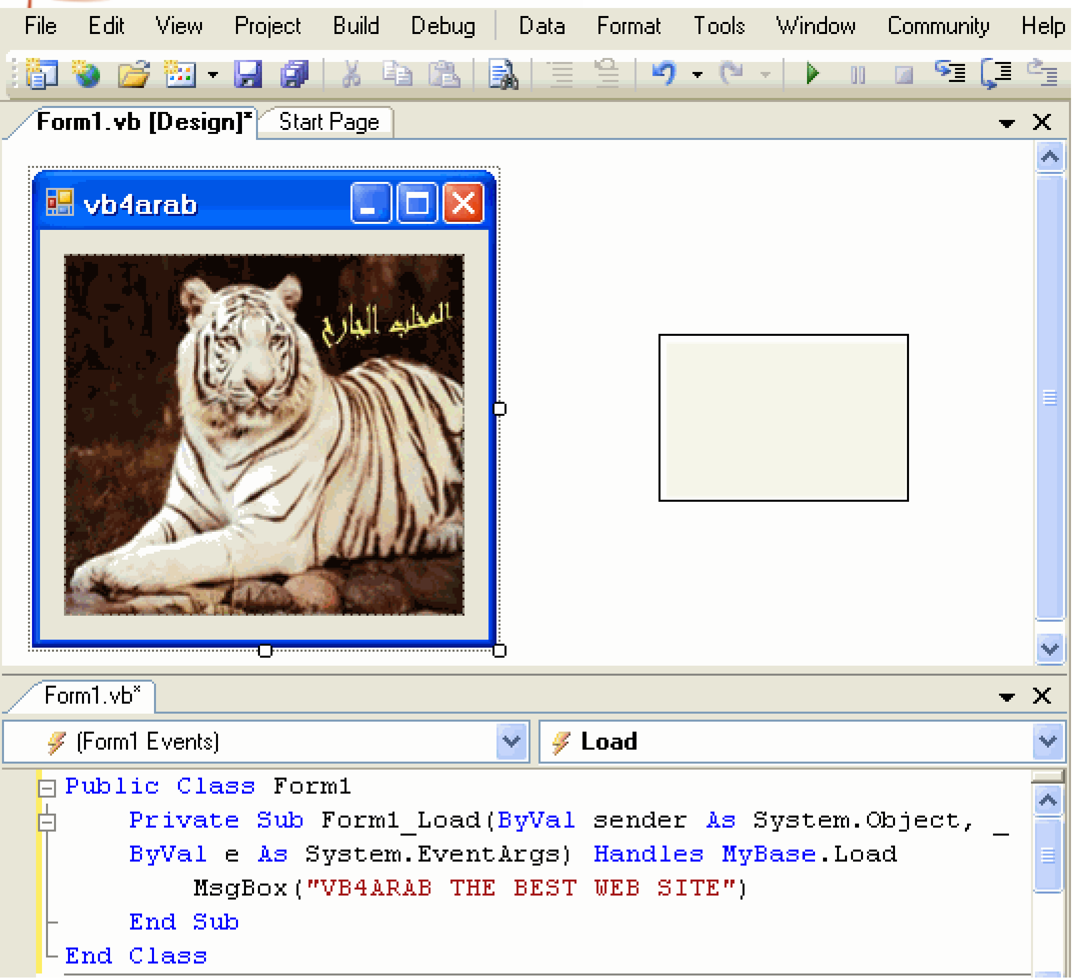
- Dockable, Floating, Tabbed Document
تتعامل مع النوافذ الثانوية مثل Toolbox, Solution Explorer، بتغيير طريقة إظهار النافذة أي تكون ظاهرة بشكل عائم أو بشكل ثابت ،أو النقر المزدوج على شريط عنوان النافذ لجعلها عائمة أو تكرار النقر لتثبيتها ،والآن قم بالنقر المزدوج على شريط العنوان لأي نافذة ثانوية كي تكون بشكل عائم ثم بواسطة الضغط المستمر بالماوس على شريط العنوان وسحب النافذة سوف تظهر لك مقابض في شاشة الفجول تستطيع بواسطة هذه المقابض تثبيت النافذة في أي مكان تريد.

- Auto Hide
تقوم بخفاء تلقائي للنافذة الثانوية وتظهر عند مرور الماوس عليه

- Hide
يقوم هذا الخيار بإغلاق النافذة الثانوية وتستطيع إظهارها مرة ثانيه من القائمة view أو من خلال الضغط على اختصار إظهارها من شريط الاختصارات

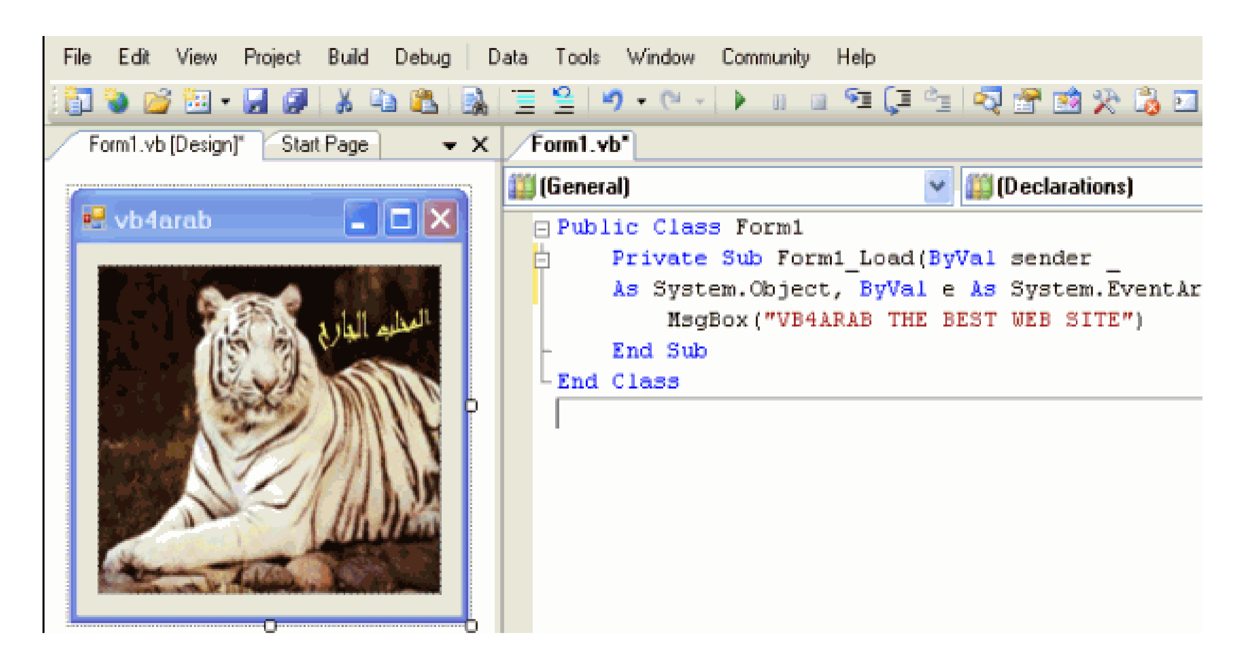
- Auto Hide All
يقوم بإخفاء كل النوافذ الثانوية الظاهرة

- New Horizontal Tab Group
يقوم هذا الخيار بعرض أكثر من محرر في وقت واحد على الشكل الأفقي أي إذا كانت تعمل على محرر التصميم للمشروع وتريد أيضا العمل على محرر الكود قم بالضغط على هذا الخيار وسيكون الشكل كما هو موضح بالصورة


File Edit View Project Build Debug Data Format Tools Window Community Help
Form1.vb [Design]*
Start Page
vb4arab
المخلب الجارح
Form1.vb*
(Form1 Events)
Load
Public Class Form1
Private Sub Form1_Load(ByVal sender As System.Object, _
ByVal e As System.EventArgs) Handles MyBase.Load
MsgBox("VB4ARAB THE BEST WEB SITE")
End Sub
End Class

- New Vertical Tab Group

يقوم هذا الخيار بعرض أكثر من محرر في وقت واحد على الشكل العمودي أي إذا كنت تعمل على محرر التصميم للمشروع وتريد أيضا العمل على محرر الكود قوم بالضغط على هذا الخيار وسيكون الشكل كما هو موضح بالصورة




```
Public Class Form1
    Private Sub Form1_Load(ByVal sender _
    As System.Object, ByVal e As System.EventAr
        MsgBox("VB4ARAB THE BEST WEB SITE")
    End Sub
End Class
```

- Close All Documents

يقوم هذا الزر بإغلاق كافة المحررات المفتوحة

- Reset Window Layout

يقوم بإعادة النافذة للوضع الافتراضي لها

- Windows

يقوم بفتح نافذة تستطيع منها تحديد أو إغلاق أي محرر مفتوح

١١- قائمة[Community]

تحتوي على الأوامر التي تساعدك في الارتباط مع مجتمع البرمجة بالفجول بيسك بحيث يمكنك أن تسأل وتقوم بالبحث عن إجابة أو أمثلة وهذه الأوامر مرتبطة بصفحات ميكروسوفت كما يمكنك زيارة MSDN Community Center من خلال هذا الرابط
http://msdn2.microsoft.com/en-us/aa497440.aspx

١٢- قائمة[Help]

تحتوي على الأوامر المألوفة الخاصة بالمساعدة والاتصال بموقع الشركة والتعرف على كيفية استخدام اللغة وخيار العثور على التحديثات اللازمة للبرنامج

الأدوات

هي عبارة عن عناصر برمجية لها مهام معينة تأخذ أشكال رسومية ولكل أداة عملها الخاص
ملاحظة: مايكروسوفت قامت بإخفاء الأدوات النادرة الاستعمال ولكي يمكنك إظهار الأداة التي تحتاجها من View ثم Toolbars واختيار الأداة التي تريد إظهارها.

*التعامل مع الأدوات:

الأدوات تساعد المبرمج أثناء تصميمه برنامج بقدر كبير فتقوم باختصار الكثير من الأكواد الصعبة والطويلة.ولكل أداة من أدوات الفجول بيسك دوت نت غرض معين ومهمة معينة وتقوم كل أداة بعمل مختلف عن الأداة الأخرى



إنشاء الأدوات:

*يمكنك إضافة أي أداة إلى برنامجك بإحدى هذه الطرق:

١- النقر المزدوج على الأداة المراد إضافتها
٢- السحب والإفلات باستخدام الماوس
٣- النقر نقرة واحدة على الأداة المراد إضافتها ثم النقر نقرة واحدة على الفورم وسيتم إضافتها

لو أردنا إنشاء مجموعة كبيرة من الأزرار كعمل برنامج حاسبة سيكون بالضغط المتواصل على مفتاح Ctrl ثم النقر نقرة واحدة باستخدام الماوس على الأداة التي تريد إضافة عدداً منها ثم النقر على الفورم ستلاحظ في كل مرة تقوم بها بالنقر على الفورم سيتم إضافة أداة جديدة و باسم جديد
.

وعندما تريد الانتهاء من إضافة الأداة فما عليك سوى إيقاف الضغط على مفتاح Ctrl .

تجميع الأدوات داخل حاوية:

هناك أدوات عملها الأساسي هو احتوى أدوات أخرى بداخلها مثل TabControl, GroupBox, Panel, وباستخدامها تستطيع تجميع عدد من الأدوات في حاوية واحدة لغرض نقل وتحريك أو إخفاء وإظهار تلك الأدوات أو لأغراض أخرى.

طرق وضع الأدوات داخل حاوية container :

- بسحب الأداة الموجود على الفورم إلى داخل container
- بتحديد ال container الذي قمت بإضافته مسبقا إلى الفورم ثم من قائمة الأدوات تقوم بالنقر المزدوج على الأداة التي تريد أن تضيفها إلى container وستضاف بداخله مباشرتا وليس على الفورم
- أو بواسطة السحب والإفلات من قائمة الأدوات إلى داخل ال container.
- أو بواسطة القص واللصق من على الفورم إلى داخل ال container.

خصائص الأدوات وكيفية التعامل معها:

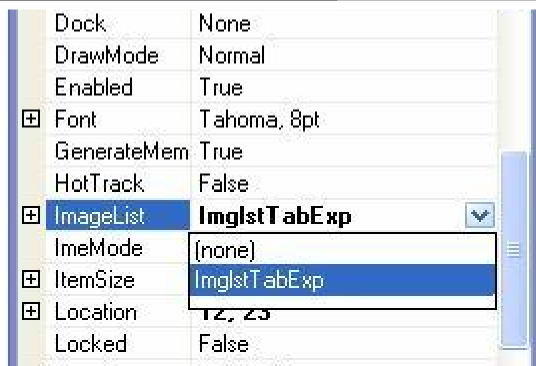
الخاصية : هي قيمة أو بعض القيم التابعة لأداة التحكم والتي تتحكم في عمل أو مظهر الأداة. مثلا أداة التحكم TextBox تمتلك الخاصية Name وهي التي تتحكم في اسم الأداة البرمجي

مثال وضع الصور بعناوين: TabControl

قم بفتح مشروع جديد بالفجول بيسك دوت نت ثم قم بإضافة هذه الأدوات إلى فورم البرنامج الأداة TabControl وعدل لخاصية Name لتكون TabExp والأداة ImageList وعدل الخاصية Name لتكون ImglstTabExp ثم قم بالنقر نقرة واحدة على الأداة ImglstTabExp فتظهر لك خصائصها بنافذة الخصائص ومنها قم بالنقر على الخاصية Images وستظهر لك نافذة محرر إضافة الصور التابع للأداة



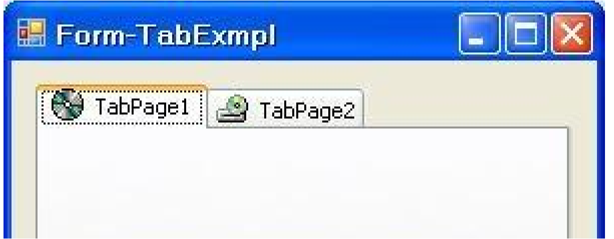
ولو لاحظت الصورة فإنه يوجد صورتين تم إضافتهما مسبقا فقم أنت الآن باختيار صورتين ونسطة الزر Add ثم بعد ذلك اضغط Ok عندها سيغلق محرر إضافة الصور وستعود إلى وضع تصميم البرنامج الآن قم بتحديد الأداة TabExp تظهر خصائصها بنافذة الخصائص والآن قم بالنقر على الخاصية ImageList التابعة للأداة TabExp وقم باختيار الأداة ImglstTabExp



والآن قم بالنقر أمام الخاصية TabPages التابعة للأداة TabExp وستظهر لك نافذة تعديل وإضافة التاب (Tabs) وباقي الشرح انظر للصورة



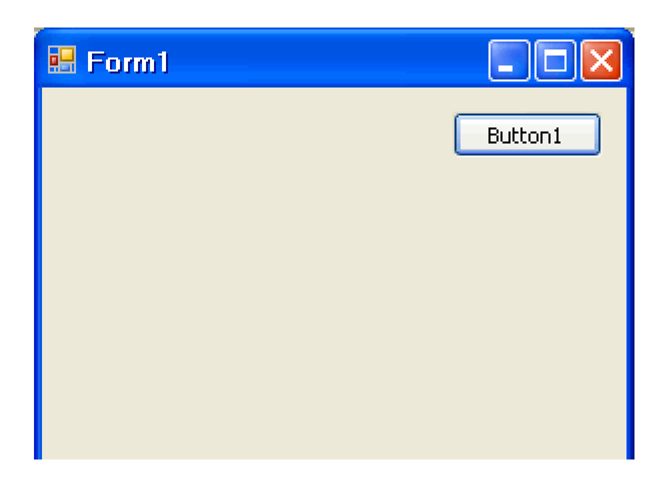
ويجب أن نعرف أن الصفحات أو ما يعرف بالتاب (Tabs) المكونة للأداة TabExp تعتبر أدوات داخلية لها خصائصها المرتكزة على خصائص الأداة الرئيسية وهي TabControl ويمكنك إضافة العدد الذي تريد من التاب من خلال الزر Add الموجود أسفل نافذة محرر إضافة وتعديل التاب وفي النهاية سوف يظهر البرنامج بهذا الشكل



الخاصيتان Anchor and Dock

الخاصيتان Anchor and Dock تعملان على جعل الأداة تقوم تلقائيا بتغيير حجمها وموضعها تبعا لحجم الفورم الموضوعة علية الخاصية Anchor تحدد أي حد من حدود الأداة يجب أن يبقى ثابتا وأي حد يجب أن يزيد عن حجمه الحالي تبعا لحجم الفورم الموضوعة علية، تملك هذه الأدوات أربع حدود هي Top, Bottom, Left, Right و تحسب قيمة هذه الحدود بالنسبة للفورم بالبكسل (pixel) والقيمة الافتراضية لهذه الخاصية هي Top, Left .

ولان لنقم بمثال قم بإضافة Button إلى الفورم وقم بوضع الزاوية العليا من الجهة اليمنى على الفورم



ثم قم بجعل قيمة الخاصية Anchor تساوي None أي قم بإلغاء تحديد كل حدود الأداة وقم بتشغيل البرنامج وتغيير حجم الفورم ستلاحظ أن الأداة تعد إلى لانقل إلى قرب منتصف الفورم والآن قم بإيقاف البرنامج وتغيير الخاصية Anchor لل Button لتكون قيمتها تساوي Top, Right ثم قم بتشغيل البرنامج وتغيير حجم الفورم ستلاحظ ثبات ال Button في الزاوية العليا من الجهة اليمنى مهما تغير حجم الفورم وذلك لأننا قمنا بتثبيت حدود الأداة Top,

Rightبواسطة الخاصية فكلما قمنا بتغيير حجم الفورم فان ال Button يعمد إلى لانتقال إلى تلك الجهة المثبتة حدوده عليها.

الخاصيةDoc

تقوم بتثبت موضع الأداة في احد جوانب الفورم أو الحاوية التي هي عليها مثلا لو قمت بتغيير الخاصية Dock لتساوي Top لأداة مثل Button فان الأداة ستنتقل إلى أعلى الفورم وتتمدد جوانبها لتصل للجانبين Left & Right. يمكنك تغيير الخاصية Dock لتساوي Top, Bottom, Left, Right, Fill, or Noneوالخاصية Fill تقوم بجعل حجم الأداة يغطي الفورم التي هي علية أو الحاوية التي تحتويها ،ولكن السؤل ماذا لو كان لدينا أكثر من أداة على الفورم قمنا بتغيير قيم الخاصيةDock لها أي أداة ستنفذ قيمتها أولا.

إنا الفورم أو الأداةستقوم بترتيب الأدوات بحسب stacking order التابع للأدوات أو ما يسمى Z-order،الأداة التي تم وضع القيمة لها أولا هي التيستأخذ موضعها أولا فالتالي والتالي.

مثل : قم بوضع أربع TextBox على الفورم وغير الخاصية MultiLine للكل لتساوي True واجعل إحجامها وموضعهن على الفورم كما بالصورة.

والآن قم بتغيير الخاصية Dock بالترتيب للكل كما يلي:

الأول : أعطيها القيمة Right

الثاني : أعطيها القيمة Top

الثالث : أعطيها القيمة Bottom

الرابع : أعطيها القيمة Fill

وستكون بعد ذلك كما بالصورة



أهم الأدوات المستخدمة فير برنامج vb.net

اسم الأداة	شكل الأداة	الوظيفة
صندوق العنوان Label		تستخدم لعرض بيانات نصية على الشاشة (غالبا ما تستخدم كعنوان لأدوات أخرى) ولا يستطيع المستخدم أن يقوم بتعديل محتوى هذه الأداة
زر الأمـــــر Button		تستخدم لوضع الأوامر التنفيذية بداخلها وعند ضغط المستخدم على هذا الزر يتم تنفيذ الأمر أو الأوامر الموجودة بداخلها
صندوق النصوص Textbox		يستخدم لعرض واستقبال البيانات من المستخدم ويمكن للمستخدم تعديل النص الموجود بداخله (إذا سمح له بذلك)
صندوق الخيارات Checkbox		يستخدم عند وضع مجموعة من الخيارات ويمكن للمستخدم اختيار أكثر من خيار
زر اختيار Radiobutton		يستخدم عند وضع مجموعة من الخيارات ويمكن للمستخدم اختيار خيار واحد فقط(وعند اختيار أحد الأزرار تزول علامة الاختيار عن باقي الأزرار)
صندوق المجموعات Groupbox		يستخدم كحاوية لبعض الأدوات التي يتم التعامل معها كمجموعة واحدة (عندما تريد تحريك مجموعة من الأدوات معا توضع داخل صندوق المجموعات)
صندوق الصور Picturebox		يستخدم كوعاء توضع الصور بداخله
أداة القوائم Mainmenu		تقوم بعمل القوائم في البرنامج
أداة التاريخ والوقت DateTimePicker		تقوم بعرض التاريخ والوقت
المؤقت Timer		يقوم بتنفيذ أمر أو مجموعة أوامر كل فترة تحدد من قبل المبرمج
صندوق حوار خط FontDialog		صندوق تنسيق الخطوط
صندوق حوار ألوان ColorDialog		يستخدم في تنسيق الألوان

المتغيرات والثوابت

تعريف:

(VARIBLES)عبارة عن مكان يتم حجزه في ذاكرة الكمبيوتر ويخصص له اسم المتغيرات ويحمل قيمة قد تتغير أثناء تنفيذ البرنامج

أنواع المتغيرات:

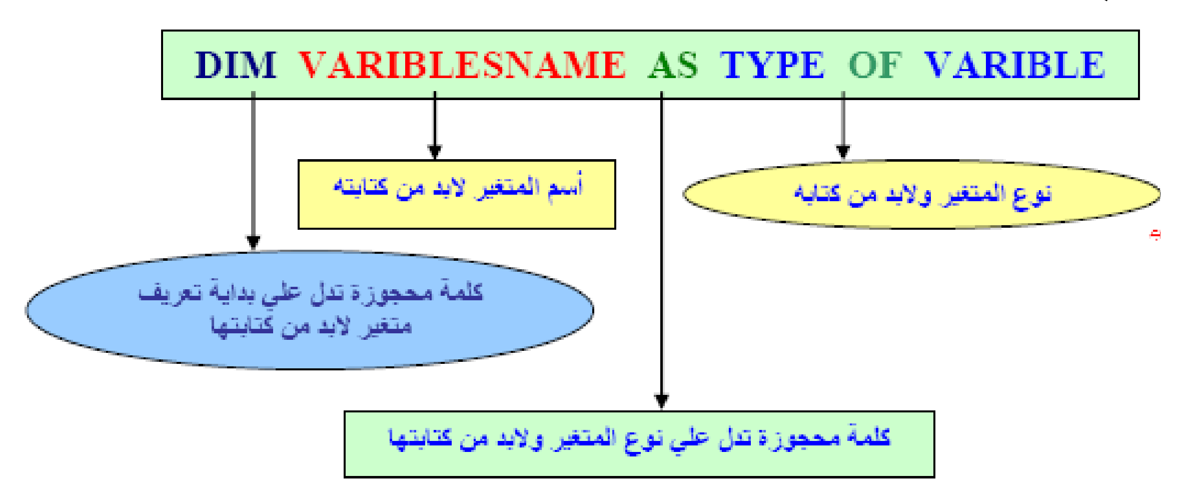
يوجد عدد كبير من أنواع المتغيرات وسوف نشاهد من خلال الجدول القادم عدد من أنواع البيانات

نوع المتغير	الشرح	حجم المتغير
INTEGER	عدد صحيح صغير نسبيا	4 BYTES
LONG	عدد صحيح طويل نسبيا	8BYTES
SINGLE	عدد حقيقي يحتوي علي فاصلة عشرية	4 BYTES
SHORT	أرقام صحيحة	2 BYTES
DOUBLE	عدد حقيقي يحتوي علي فاصلة عشرية (كبير نسبيا)	8 BYTES
DECIMAL	عدد ذو علامة عشرية يحتوي علي ١٥ رقم صحيح و ٤ أرقام عشرية	8 BYTES
STRING	سلسلة من الحروف ثابتة الطول	يعتمد علي الطول
BOOLEAN	قيمة منطقية	2 BYTES
DATE	التاريخ والوقت	8 BYTES
CHAR	يحتوي علي حرف واحد فقط	2 BYTES

لقد ذكرنا من خلال الجدول السابق بعض أنواع البيانات وإلا كثر استخدام وشيوعا

كيفية كتابة المتغيرات داخل الفجول بيسك

الصيغة العامة:

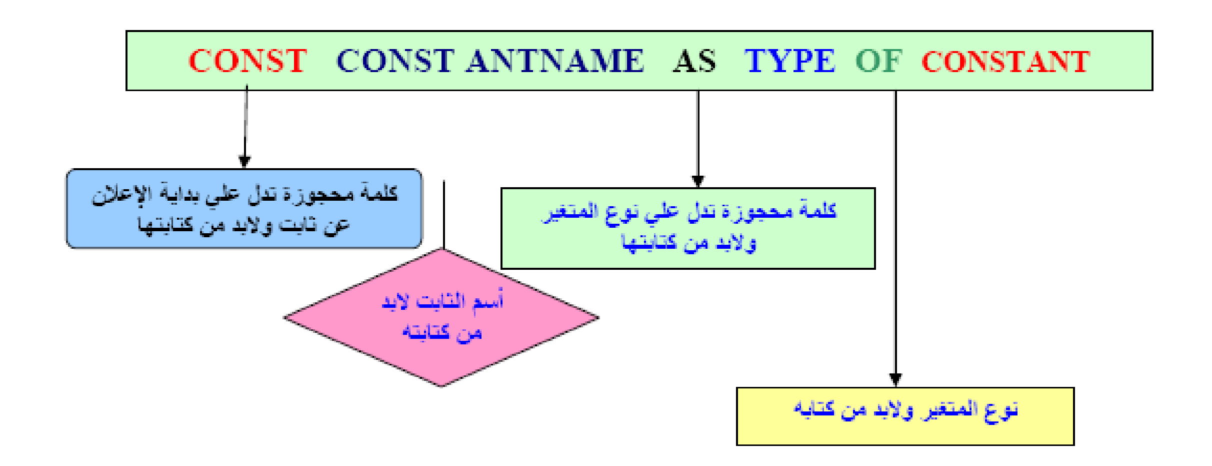


كيفية كتابة الثوابت داخل الفجول بيسك دوت نت.

تعريف:

(CONSTANT)عبارة عن أسم يحمل قيمة ثابتة لا تتغير أثناء تنفيذ التطبيق .

الصيغة العامة





ملاحظات:

- إن المتغيرات في بيئة الدوت نت ليست مجرد أسماء لتخزين القيم فحسب، بل هي عبارة عن كائنات objects وإجراء العمليات على المتغيرات. وهذا هو ما يميز بيئة الدوت نت ولغتنا تحديدا
- رابعة ل VB.NET غير حساسة Case-insensitive من ناحية الكتابة بالأحرف الكبيرة أو الصغيرة. فمثلا كتبت INTEGER أم integer فكلاهما سواسية، حيث أن المحرر سيقوم بتحويلها إلى الشكل القياسي وهو أن تبدأ الكلمة بحرف كبير وبقية الحروف تكتب صغيرة.

شروط كتابة المتغيرات:

- ألا يكون المتغير من الكلمات المحجوزة(التي تظهر باللون الأزرق في المحرر)
- ألا يزيد عن ٢٥٥ حرفا.
- ألا يتخلله مسافات، ويمكن استخدام رمز الشرطة السفلية (_) عوضا عن المسافات.
- ألا يبدأ بأرقام،ولكن يمكن أن تتخلله أو أن ينتهي بأرقام.
- ألا يتم تعريف متغيرين بنفس الاسم في نفس التطبيق.

الإعلان المتعدد للمتغيرات:

الإعلان المتعدد يعني إمكانية الإعلان عن أكثر من متغير في نفس الجملة البرمجية

فمثلا بدل أن نكتب

```
Dim x As Integer
Dim y As Integer
```

نكتبها بهذا الشكل

```
Dim x , y As Integer
```

وبدل أن نكتب

```
Dim x As Integer
Dim s As String
```

نكتبها بهذا الشكل

```
Dim x As Integer, s As String
Dim x , y As Integer , s As String
```

كما يمكن إسناد قيم أولية

```
Dim x As Integer =10, s As String="MGR",z As Float =10.5
```

العمليات الحسابية

() يقوم البرنامج بحساب ما بين الأقواس.
^ يقوم البرنامج بحساب الأس .
* يقوم البرنامج بحساب الضرب.
\ يقوم البرنامج بعملية القسمة بدون احتساب كسور في النتيجة.
Mod يقوم البرنامج باحتساب باقي القسمة.
+- يقوم البرنامج بعمليتي الجمع والطرح.



:ملاحظة

ويجب أن ألفت انتباهك إلى أهمّيّة وضع الأقواس في العمليّات المتداخلة، وذلك حتّى تضمن صحّة إجراء العمليّة بالترتيب الذي تريدها به. حيث:

١- يتمّ تنفيذ ما بين الأقواس أولا.
٢- إذا لم تكن هناك أقواس يتمّ تنفيذ الأسس أولا.
٣- ثمّ يتمّ تنفيذ الضرب والقسمة.
٤- ثمّ بعد ذلك يتمّ تنفيذ الجمع والطرح.

الطريقة التقليديّة : الطريقة المختصرة المكافئة

X = X + 1
X + = 1

X = X + Y
X += Y

X = X – 4
X -= 4

X = X * 2
X *= 2

X = X / 9
X /= 9

X = X ^ 3
X ^= 3

ولك الحرية في اختيار الطريقة التي تريد.

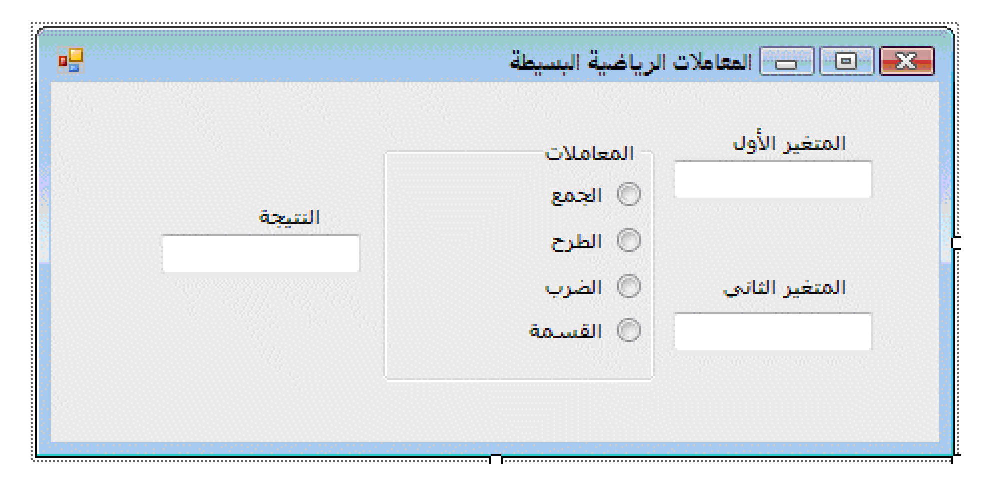
مثـــال المعـــاملات الرياضـــية البـــسيطة + :، -، *،/
المعاملات التابعة للجمع أو الطرح أو الضرب أو القسمة هي معاملات رياضية بسيطة التعامل وقد تعرفنا عليها في دراستنا الابتدائية في المدرسة، هنا سنقوم فقط بأخذ مثال برمجي تطبيقي لنرى كيف نستفيد من هذه المعاملات .
قم بإنشاء مشروع جديد وليكن اسمه Basic Math ثم أضف المكونات التالية للفورم:

1-عدد ثلاثة صناديق نص TextBox
2-واحد صندوق المجموعات GroupBox
3-أربعة أزرار راديو Radio Button إلى صندوق المجموعات
4-ثلاثة ليبلات Labels

قم بضبط الخصائص كالتالي:

١- الفورم:
RightToLeft = Yes
Text = المعاملات الرياضية البسيطة

٢- لليبلات:
نغير خاصية الـ Text للأول "المتغير الأول"، وللثاني "المتغير الثاني"، وللثالث "النتيجة"
ونغير خاصية الـ Text لصندوق المجموعات إلى المعاملات.
وكذلك نغير خاصية الـ Text لأزرار الراديو كالتالي: الأول "الجمع"، الثاني "الطرح"، الثالث "الضرب"، الرابع "القسمة". بعد التصميم وتعديل الخصائص سيكون الفورم مثل الشكل التالي:



نضغط Double-Click على زر الراديو "الجمع" لنذهب إلى منطقة الكود لنكتب الأكواد التالية:
رمز:

```
Dim x, y As Double
X = TextBox1.Text
Y = TextBox2.Text
TextBox3.Text = x + y
```

وكذلك نكتب الكود التالي لزر الراديو الطرح:

رمز:

```
Dim x, y As Double
x = TextBox1.Text
y = TextBox2.Text
TextBox3.Text = x - y
```

وهذا للضرب:

رمز:

```
Dim x, y As Double
x = TextBox1.Text
y = TextBox2.Text
TextBox3.Text = x * y
```

وهذا للقسمة:

رمز:

```
Dim x, y As Double
x = TextBox1.Text
y = TextBox2.Text
TextBox3.Text = x / y
```

نقوم بتشغيل البرنامج ثم نقوم بإضافة أرقام إلى صندوق النص الأول والى الثاني ثم جرب العمليات الحسابية (الجمع والطرح والضرب والقسمة).

مثل لاستخدام المعاملات الرياضية المتقدمة:

سقوم باستخدام المثال الأول مع تغيير خاصية الـ Text لأزرار الراديو كالتالي:

الزر الأول من "الجمع" إلى "القسمة بدون إظهار الكسور"

الزر الثاني من "الطرح" إلى "باقي القسمة"

الزر الثالث من "الضرب" إلى " الأس"

الزر الرابع من "القسمة" إلى "ضم الكلمات"

ثم نقوم بتغيير الأكواد التابعة لكل زر فنبدل المعاملات التالية فقط:

من المعامل + إلى\

من المعامل – إلىMod

من المعامل * إلى^

من المعامل / إلى&

بعد التعديلات السابقة، قم بتشغيل البرنامج سيكون كما في الشاشة التالية:

خطأ!

المعاملات الرياضية البسيطة

المتغير الأول
47
المعاملات
القسمة بدون إظهار الكسور
باقي القسمة
الأس
ضم الكلمات
النتيجة
4
المتغير الثاني
10

لاحظ أن "القسمة بدون إظهار الكسور" **للرقم ٤٧ على الرقم** ١٠ يساوي ٤ **كما في الصورة** أعلاه، وتفيدنا هذه العملية في حساب الأشياء التي لا تقبل التقسيم إلى مكونات اصغر فمثلا يمكن حساب عدد الأشخاص الذين تستطيع حملهم السيارة **بواسطة المعامل أعلاه**. (فلا يمكن أن نقول حمولة السيارة = ٤٫٧ **أشخاص**)، **وكذلك عدد** السيارات التي قد تحملها السفينة الواحدة، وغيرها من العمليات التي لا تقبل التقسيم **إلى مكونات اصغر**. للعلم التطبيق أعلاه موجود بالمرفقات **رقم** ١٥٠.

المعاملات العلاقية

أمثلة على المعاملات العلاقية أعلاه:

المعامل	اسمه
=	يساوي
>	أكبر من
<	أصغر من
>=	أكبر من أو يساوي
<=	أصغر من أو يساوي

رمز:

```
Price < 100
```

لكي تكون الجملة الشرطية أعلاه صحيحة فلا بد أن يكون السعر **أقل من مائة وإذا كان أكثر** فالجملة الشرطية خاطئة، ماذا إذا كان السعر يساوي مائة ستظل الجملة الشرطية خاطئة **لأننا حددنا في** الجملة الشرطية بأن السعر لابد وأن يكون أقل من مائة، أما إذا **كانت** الجملة الشرطية كالتالي:

رمز:

```
Price <= 100
```

فإن السعر إذا كان مائة أو أقل فإن الجملة الشرطية صحيحة **(أو** بالأصح نتيجة الجملة الشرطية صحيحة). لننظر جملة شرطية **أخرى**:

رمز:

```
Tall <> 150
```

فإذا كان الطول Tall يساوي ١٥٠ فإن نتيجة الجملة خاطئة أما إذا كان لا يساوي ١٥٠ **فإن** نتيجة الجملة الشرطية صحيحة. لنرى كيف يمكن الاستفادة من هذه **الشروط كالتالي** :

رمز:

```
Score < 20
Score = Label1.Text
```

فإذا كان الرقم الموجود في Label1 **أصغر من** ٢٠ **فإن** نتيجة الجملة الشرطية **صح وإذا كان الرقم الموجود في** Label1 أكبر أو يساوي ٢٠ **أو كان** عبارة عن نص فإن نتيجة الجملة الشرطية خاطئة .

المعاملات المنطقية Logical Operators

هي معاملات تتيح لنا الجمع بين اثنين أو أكثر من المعاملات الشرطية وهذه المعاملات تستخدم كلمات أساسية Keywords وليس رموزاً.

المعامل	الحالة
AND	يجب أن يكون كلا الجانبين صحيحا
OR	يجب أن يكون أحد الجانبين صحيحا أو كلا الجانبين صحيحا
NOT	ينقض القيمة الصحيحة

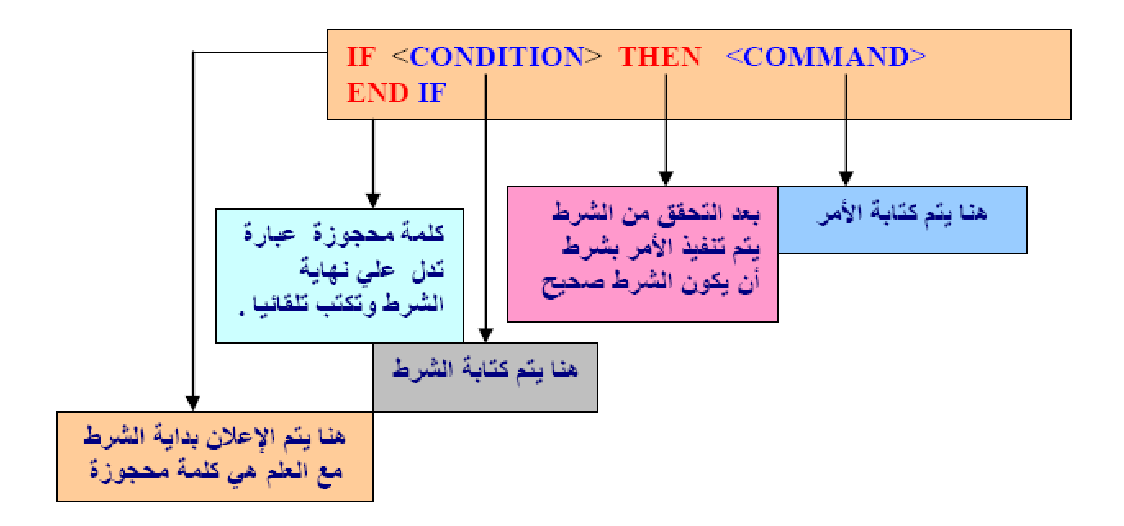


ملاحظة:

عندما تستخدم العديد من المعاملات **المترابطة** في الجملة البرمجية فإن الكمبيوتر يقوم بالتعامل **أولاً مع المعاملات** الرياضية (+، وغيرها) ثم يقوم بالتعامل مع معاملات المقارنة(<، > ، =، وغيرها)، وثالثاً يقوم بتحليل المعاملات المنطقية(And, Or) ، وغيرهما

جملة إذا كانت ... فـ If Then Decision Structure

إذا كان لدينا شرط ما إذا تحقق فإن برنامجنا يقوم بتنفيذ أمر ما فإننا نستخدم أداة الشرط إذا كالتالي



رمز:

```
إذا كان (الشرط) فقم بالتالي
If condition then statement
```

مثال:

رمز:

```
If Score >= 20 Label1.Text = "You Win"
```

فإذا كانت الدرجةٍ Scoreتساوي ٢٠ أو أكثر (فقط) ستظهر الجملة "You Win" في Label1، أي سيقوم فيجوال بيسك بتغيير الخاصية Text التابعة للمكون Label1 إلى الجملة الموضحة غير ذلك لن يتم تغيير الخاصية Text للمكون المذكور.

جملة If الشرطية التي تعالج أكثر من شرط:

نستطيع معالجة أكثر من شرط باستخدام أداة الشرط If ، لدينا المثال التالي:

```
IF <CONDITION> THEN <COMMAND>
ELSE
<COMMAND>
END IF
```

رمز:

```
If Result < 50 Then
    MsgBox("راسب")
  ElseIf Result >= 50 Then
    MsgBox("ناجح")
```

كما هو واضح في المثال أعلاه بأننا نستخدم ElseIf لإضافة شرط جديد إلى الجمل الشرطية، فإذا كانت النتيجة أقل من ٥٠ درجة فإن الطالب راسب وإذا كانت أكثر من خمسين درجة فإن الطالب ناجح ويكون تنفيذ الأمر بإظهار صندوق حوار Message Box فيه كلمة ناجح أو راسب. ماذا إذا كنا نريد من التطبيق إظهار تقدير للطالب(مقبول، جيد، جيد جداً، ممتاز)، في مثل هذه الحالة سنحتاج للكثير من الجمل الشرطية المترابطة فيما بينها كما في المثال التالي:

رمز:

```
Dim Result As Single = 71
If Result < 50 Then
MsgBox("راسب")
ElseIf Result <= 60 Then
MsgBox("مقبول")
ElseIf Result <= 70 Then
MsgBox("مقبول مرتفع")
ElseIf Result <= 80 Then
MsgBox("جيد جداً")
ElseIf Result <= 90 Then
MsgBox("جيد جداً مرتفع")
ElseIf Result <= 100 Then
MsgBox("ممتاز")
Else
MsgBox("هناك خطأ ما، تأكد من المدخلات")
End If
```

عند تنفيذ البرنامج سيقوم الكمبيوتر بالتأكد من قيمة النتيجة Result فإذا كانت أصغر من ٥٠ سيظهر صندوق حوار يفيدنا بأن الطالب راسب ولكن في حالة ما نتيجة الطالب ناجح ولكن أصغر من ٦٠ سيظهر صندوق حوار يفيد بأن الطالب مقبول، (للتأكد فقط يمكننا تغيير قيمة Result الموجودة في أول سطر) وهكذا مع بقية الشروط أو الحالات التي حددناها في الأسطر البرمجية أعلاه من جيد إلى ممتاز، ما يهمنا الآن هو الشرط الأخير Else وما بعدها فهذا الأمر أو هذه الحالة تعني أنه إذا لم تطابق أي حالة من الحالات أعلاه فقم بتنفيذ هذا الأمر وتفيدنا Else كثيراً إذا كان لدينا العديد من الخيارات الخاطئة وخيار واحد فقط هو الصحيح (أو العكس)،

فبإمكاننا كتابة الأمر If للخيار الوحيد واستخدام Else لجميع الخيارات الأخرى

ملاحظة هامة:

الترتيب في جملة If الشرطية هام جداً، لان الكمبيوتر يتوقف عن البحث عن أقرب الشروط المطابقة لوجود أول شرط مطابق(وعليه فلا بد أن يكون الترتيب بشكل تصاعدي للجمل الشرطية

عبارة if المتداخلة

```
IF <CONDITION> THEN <COMMAND>
ELSEIF
<COMMAND>
ELSE
<COMMAND>
END IF
```

استخدام Select Case

باستخدام الأداة الشرطية Select Case والتي تعني "اختر الحالة"، نعطي للكود البرمجي سهولة أكثر في القراءة والمراجعة، وعندما يكون لدينا قائمة List Box أو Combo Box تفيدنا كثيراً إذا كان لدينا متغير واحد وبخيارات متعددة. الهيئة العامة لـ Select Case كالتالي:رمز:

```
Select Case Variable
Case Values1
Statment1
Case Value2
Statment2
Case Value3
Statment3
End Select
```

مثال : فلنأخذ هذا المثال وهو عبارة عن ترجمة لما تعلمناه عن الأداة الشرطية Select Case: لنفرض أن المستخدم سيقوم بإدخال العمر (عمر المستخدم) ، فإذا أدخل المستخدم العمر ١٢ نريد أن تظهر له رسالة تقول له "أنت قاصر بحسب القانون"، أما إذا أدخل العمر ١٥ فتظهر له رسالة تقول له "أنت مكلّف بحسب القانون المدني" ، وإذا أدخل العمر ١٨ تظهر له رسالة تقول له" الآن يحق لك امتلاك وإدارة الشركات"، وإذا أدخل العمر ٤٠ تظهر له رسالة تقول له "دخلت مرحلة الوقار" وإذا أدخل العمر ٦٥ تظهر الرسالة " بدأت عمر التقاعد". لنقوم بتطبيق المثال أعلاه:-

رمز:

```
Dim Age As Integer
Age = 18
Select Age
   Case 12
      MsgBox("أنت قاصر بحسب القانون")
   Case 15
      MsgBox("أنت مكلف بحسب القانون المدني")
   Case 18
      MsgBox("الآن يحق لك امتلاك وإدارة الشركات")
   Case 40
      MsgBox("دخلت مرحلة الوقار")
   Case 65
      MsgBox("بلغت عمر التقاعد")
End Select
```

لاحظ بأننا حددنا العمر بـ ١٨ في بداية الكود لكنه في نفس الوقت بإمكاننا جعل العمر في TextBox قابل للتغيير من قبل المستخدم وسيقوم البرنامج بإظهار رسائل الحوار تبعاً للأرقام التي سوف يكتبها المستخدم.

كما تدعم أداة الشرط Select Case تستخدم Case Else قبل كتابة End Select، فبإمكاننا كتابة الجملة البرمجية كاملة بـ Select Case وإذا لم تطابق أي من المدخلات من قبل المستخدم فإنه سيعمل بالأمر البرمجي الموجود بعد Case Else ومعناها (أي حالة أُخرى غير مذكورة).
وبإمكاننا كذلك استخدام المعاملات المنطقية (=, <>, <, >, <=, >=) في الجمل البرمجية الشرطية بـ Select Case ولكن بعد إضافة Is بعد كلمة Case. نطبق بالمثال التالي:

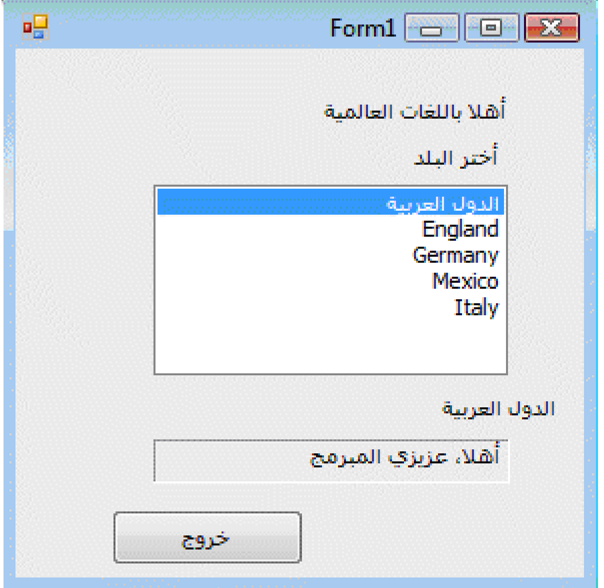
رمز:

```
Dim Age As Integer
Age = 66
Select Case Age
   Case Is < 15
      MsgBox("أنت قاصر بحسب القانون")
   Case 15
      MsgBox("أنت مكلف بحسب القانون المدني")
   Case 18
      MsgBox("الآن يحق لك امتلاك وإدارة الشركات")
   Case 40
      MsgBox("دخلت مرحلة الوقار")
   Case Is >= 65
      MsgBox("بلغت عمر التقاعد")
   Case Else
      MsgBox("تأكد من الرقم المدخل، هناك خطأ ما")
End Select
```

نستخدم Select Case إذا كان لدينا متغير واحد وله ثلاث أو أكثر من ثلاث حالات أو قيم، أما إذا كان لدينا متغير وله حالتين أو أقل فيفضل استخدام. If..Then

مثل:

أنشئ مشروعاً جديداً وسمه My Select Case ، قم بإضافة أربعة ليبلات و , ListBox و زر Button إلى الفورم ثم قم بتعديل الخصائص لتتناسب مع الشكل أدناه



نضغط Double-Click على الفورم فيحولنا إلى صفحة الكود وبالتحديد إلى الحدث Load التابع للفورم، نكتب فيه الكود التالي، لتحميل العناصر في قائمة:ListBox

رمز:

```
ListBox1.Items.Add("الدول العربية")
ListBox1.Items.Add("England")
ListBox1.Items.Add("Germany")
ListBox1.Items.Add("Mexico")
ListBox1.Items.Add("Italy")
```

ثم نضغط على الزر ونكتب الكود التالي لإغلاق البرنامج:

رمز:

```
End
```

وكذلك نضغط على القائمة ListBox لنتحول مباشرة إلى منطقة الكود وبالضبط إلى الحدث SelectedIndexChanged التابع للقائمة ListBox1 ونكتب فيها هذا الكود الذي يحدد ماذا اختار المستخدم من بنود القائمة ويكتب جملة الترحيب في Label4 وكذلك يكتب نص الاختيار في:Label3

رمز:

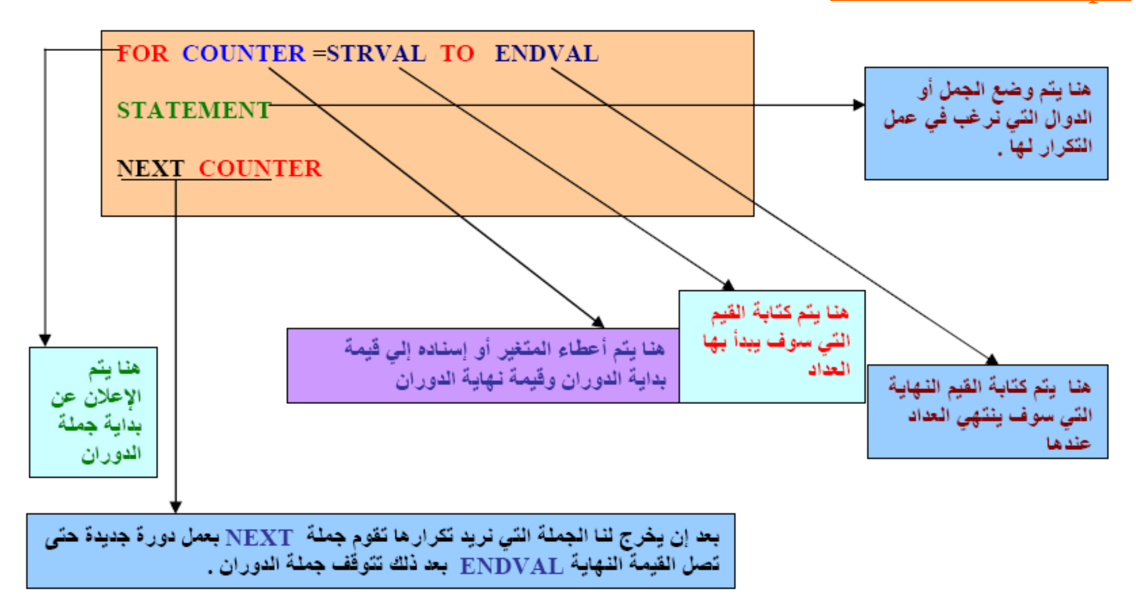
```
Select Case ListBox1.SelectedIndex
   Case 0
      Label3.Text = ListBox1.SelectedItem
      Label4.Text = "أهلا، عزيزي المبرمج"
   Case 1
      Label3.Text = ListBox1.SelectedItem
      Label4.Text = "Hello, programmer"
   Case 2
      Label3.Text = ListBox1.SelectedItem
      Label4.Text = "Hallo, programmierer"
   Case 3
      Label3.Text = ListBox1.SelectedItem
      Label4.Text = "Hola, programador"
   Case 4
      Label3.Text = ListBox1.SelectedItem
      Label4.Text = "Ciao, programmatore"
End Select
```

قم بتشغيل البرنامج وجرب أختر أية بند من بنود القائمة ListBox ماذا تلاحظ، ستلاحظ انتقال اسم البند إلى Label3 وكذلك ظهور جملة في Label4 بحسب الجُمل التي أضفناها في منطقة الكود.ستجد نسخة من المشروع أعلاه مع المرفقات برقم ١٨٠. الآن انتهينا من الجمل البرمجية الشرطية باستخدام If..Then وكذلك باستخدام Select Case

الحلقات التكرارية : Loop

نستخدم الحلقات التكرارية لتكرار سطر او امر عدة مرات ، سنشرح طرق التكرار في هذا الدرس:

For - Next Loop :



حلقة تكرار يتم تحديد نقطة البداية والنهاية ومقدار الخطوة فيها ، ابسط مثال عليها الكود التالي:

VB.net:

رمز:

```
For i As Integer = 0 To 9

   Console.WriteLine(i)
Next
```

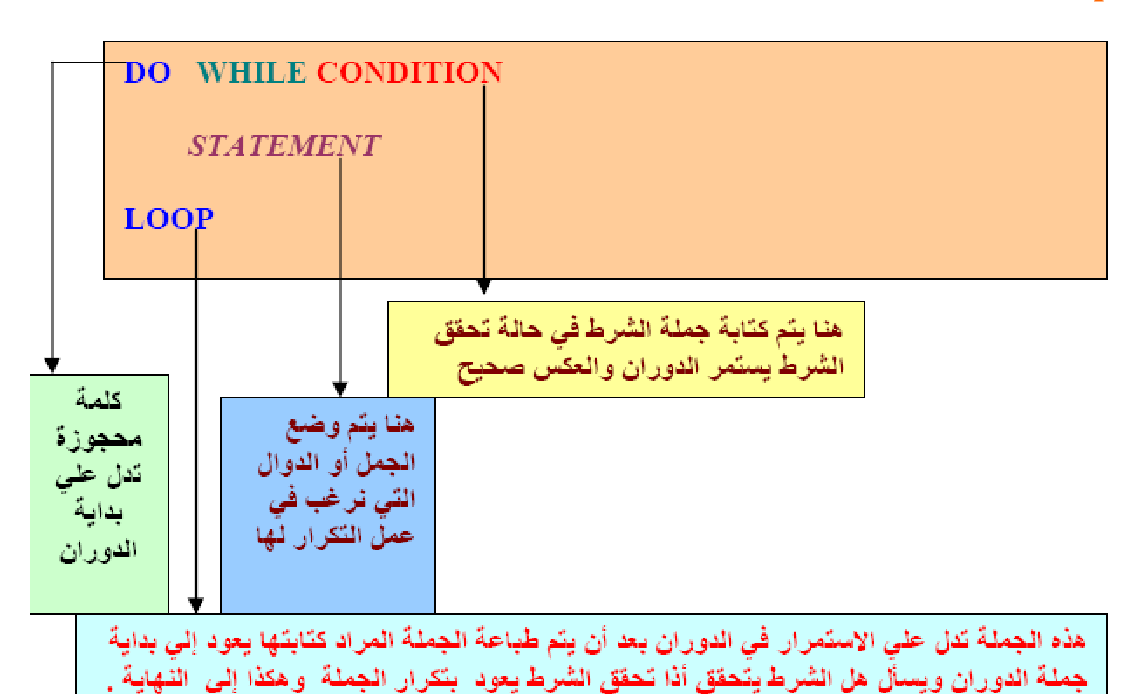
يمكن عمل مقدار القفز = ٢ مثلاً **بالشكل التالي** - لطباعة الارقام الزوجية مثلاً: -

رمز:

```
For i As Integer = 0 To 9 Step 2

   Console.WriteLine(i)
Next
```

While Loop :



الطريقة الثانية لكتابة الجملة

```
DO
    STATEMENT
LOOP  WHILE CONDITION
```

الفرق بين الطريقة الأولى والثانية

الطريقة الاولي	الطريقة الثانية
في الصيغة الاولي يتم أختبار الشرط قبل عملية الدوران بمعني لن تتم عملية الدوران الا اذا تحقق الشرط .	في الصيغة الثانية تتم عملية الدوران حتي يصل الشرط ففي حالة تحقق الشرط يستمر في الدوران واذا لم يتم الشرط فأنة سوف ينفذ العملية مرة واحدة.

أمثلة : حلقة تكرار تستمر في العمل حتى تحقق شرط معين ، لنفترض مثلاً حتى وصول العداد إلى رقم ١٠:

رمز:

```
Dim x As Integer = 0
While x < 10

   Console.WriteLine(x)
End While
```

حالة اخرى غير رقمية ، ادخال بيانات حتى ادخل exit ، في هذه الحالة سوف نضيف Do في البداية ونضع While في النهاية.

رمز:

```
Dim inp As String
Do
   inp = Console.ReadLine()

   Console.WriteLine(inp)
Loop While inp <> "exit"
```

For Each Loop

للدوران على (كل) شيء معين ، مثلاً للدوران على كل عناصر مصفوفة معينة مثلاً:

رمز:

```
Dim arr As Integer() = {10, 20, 30, 40}
For Each i As Integer In arr
   Console.WriteLine(i)

Next
```

مصطلحات:.

الجمل البرمجية:(Program Statements)

الجمل البرمجية هي عبارة عن الجمل المكتوبة في السطور البرمجية (خانة الكود) وتقوم هذه الجمل بعمل ما خلال مرحلة تنفيذ البرنامج لان .

الكلمات المحجوزة:(Keywords)

الكلمات المحجوزة keywords هي كلمات محجوزة في بيئة التطوير هذه الكلمات تتعامل مع الكومبايلر بالطريقة التي قد حددت سلفاً من قبل مطوري لغة البرمجة (مايكروسوفت). مثل الكلمة end وتستخدم لإغلاق البرنامج أو التطبيق وعليه فلا يمكنك أن تقوم بتعريف متغير بنفس الكلمة. الكلمات المحجوزة تعتبر جزء من بنية الجمل البرمجية التابعة للفيجوال بيسك معظم الكلمات المحجوزة تظهر باللون الأزرق في محرر الكود .

المتغيرات:(Variables)

المتغيرات هي عبارة عن حاضنات أو حافظات للبيانات تحفظ البيانات بشكل مؤقت ويتم تعريف المتغيرات باستخدام كلمة Dim قبل المتغير وتقوم هذه المتغيرات بحفظ البيانات بشكل مؤقت .

أدوات:(Controls)

الأدوات هي عبارة عن تلك الأدوات التي تقوم بإضافتها إلى الفورم مثل الأزرار، صناديق النص وصناديق الصور وغير من الأدوات التي يمكنك إضافتها إلى الفورم (عادة ما تكون فغي واجهة المستخدم.

الكائنات:(Objects)

الكائنات هي عبارة عن عناصر التي تقوم بصنعها بواسطة برنامج الفيجوال بيسك باستخدام أحد الأدوات الموجودة في صندوق الأدوات .Toolbox في نفس الوقت الكائنات قد تكون مقدمة من ملحقات نظام التشغيل والعديد من هذه الكائنات تحتوي على بيانات.

خلاصة الكتاب

من اجل أن تقم بالتالي

تفتح الفيجوال ٢٠٠٨ : Start > All Programs > Microsoft Visual Studio 2008 > ثم اختر الأيقونة Microsoft Visual Studio 2008.

فتح مشروع موجود مسبقا: بعد فتح الفيجوال ٢٠٠٨ تحت قائمة File اختر Open Project أو قد تجد اسم المشروع على القائمة الرئيسية لآخر المشاريع المفتوحة فاختاره.

تنفيذ البرنامج تحويله إلى برنامج exe: من قائمة Debug اختر Start Debugging أو ضغط على F5

تغيير الخصائص: اختر المادة المراد تغيير خصائصها (حددها على الفورم) ثم اذهب إلى نافذة الخصائص واختر الخاصية التي تريد أن تغيرها ثم غيرها.

التنقل بين الملفات المفتوحة : بواسطة الضغط على Ctrl+Tab ثم انتقل بينها بتكرار الضغط على Tab أو تنقل بالأسهم بين الملفات المفتوحة وأدوات التطوير. وتستطيع اختار المادة المعنية بواسطة الماوس خلال الضغط على Ctrl+Tab

التنقل بين أدوات التطوير : بالضغط على Alt+F7 للتنقل بينها أو Alt+Shift+F7 للتنقل بطريقة عكسية.

تعديل طريقة الحصول على التعليمات: من قائمة Tools اختر Options

تعديل بيئة التطوير لتناسب العمل على فيجوال بيسك : من قائمة Tools اختر Import and Export Settings ثم اختر Reset All Settings ثم التالي ثم Save my current settings ثم التالي. أخيرا اختر Visual Basic Developments Settings وثم زر Finish ثم اختر إغلاق.

تعديل خيارات بيئة التطوير : من قائمة Tools اختر Options ثم قم بتعديل خيارات بيئة التطوير على حسب المجموعات المذكورة (لتعديل خيارات المشاريع اذهب إلى General تحت خيار Projects And Solutions لتعديل خيارات المترجم إلى لغة الآلة الـCompiler اذهب إلى Defaults VB تحت نفس المجموعة)

إغلاق الفيجوال 2008: تحت قائمة File اختر Exit


تمت